AD-A039 853 INTEGRATED SCIENCES CORP SANTA MONICA CALIF F/G 12/2
EXPERIMENTAL INVESTIGATION OF SKETCH MODEL ACCURACY AND USEFULN--ETC(U)
MAY 77 G W IRVING, J J HORINEK, P Y CHAN N00014-75-C-0811
UNCLASSIFIED ISC-215-3 NL

1 OF 2
AD
A039853

(12)

EXPERIMENTAL INVESTIGATION OF SKETCH MODEL ACCURACY AND USEFULNESS IN A SIMULATED TACTICAL DECISION AIDING TASK

REPORT NO. 215-3

Prepared for:

DIRECTOR, ENGINEERING PSYCHOLOGY PROGRAM
CODE 455
PSYCHOLOGICAL SCIENCES DIVISION
OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
ARLINGTON, VIRGINIA 22217

MAY 1977

INTEGRATED SCIENCES CORPORATION
Santa Monica, California

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle) EXPERIMENTAL INVESTIGATION OF SKETCH MODEL ACCURACY AND USEFULNESS IN A SIMULATED TACTICAL DECISION AIDING TASK.

5. TYPE OF REPORT & PERIOD COVERED
TECHNICAL REPORT

6. PERFORMING ORG. REPORT NUMBER
ISC-215-3

7. AUTHOR(s)
W. IRVING, Pak Y. CHAN
J. HORINEK, David H. WALSH

8. CONTRACT OR GRANT NUMBER(s)
N00014-75-C-0811

9. PERFORMING ORGANIZATION NAME AND ADDRESS
INTEGRATED SCIENCES CORPORATION
1640 Fifth Street, Suite 204
Santa Monica, California 90401

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
ENGINEERING PSYCHOLOGY PROGRAM, CODE 455
OFFICE OF NAVAL RESEARCH
ARLINGTON, VIRGINIA 22217

12. REPORT DATE
MAY 1977

13. NUMBER OF PAGES
182 p.

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)

UNLIMITED

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Reproduction in whole or in part is permitted for any purpose of the United States Government.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Decision Aids
Naval Tactics
Mission Planning
Interactive Graphics
Man-Computer Functional Allocation
Sketch Model
Dynamic Programming Optimization
Subjective Probability
Command and Control

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the implementation and experimental evaluation of a proposed decision aid to support naval Task Force Commanders. The decision aid, named the "Sketch Model," enables a human operator to specify his subjective estimate of a multidimensional, multimodal, unsymmetric three dimensional function by electronically "sketching" its two-dimensional projection (e.g., iso-altitude contours of a three dimensional function) on a computer graphics terminal. The Sketch Model is hypothesized to have certain advantages over comparable

→ next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

404569 EHL

20. Abstract (Continued)

decision aiding techniques (such as theoretical and empirical objective modeling and scalar subjective judgment). These advantages include ease, speed, and accuracy of estimation and of updating; consideration of qualitative data; and the capability of Bayesian estimation without explicit *a priori* data.

For the experimental evaluation, the Sketch Model technique was applied to the tactical planning problem of optimizing an air strike path. The goodness of the path was measured by a utility function which considered the cumulative probability of being detected by enemy sensors along the path and the fuel consumed along the path. A multi-sensor composite detection rate surface was Sketch Modeled via a computer graphics display and proposed strike paths were defined and evaluated. The experiment consisted of collecting data on path utilities generated by four different methods: (1) operator-performed path optimization, unaided by Sketch Models; (2) operator-performed optimization aided by operator-generated Sketch Models; (3) computerized optimization aided by operator's Sketch Models; (4) computerized optimization aided by computer-generated (or "true") composite detection rate surface.

After analysis, the data indicated that human operators were able to produce accurate (as measured by a percent volume error metric) Sketch Models of highly irregular multi-sensor detection rate fields. The data also suggested that under appropriate conditions, automated optimization supported by operator-performed Sketch Modeling can contribute to improved performance over current procedures. However, the difficulty of the optimization portion of the experimental problems was not sufficient to provide a good test of this hypothesis.

It is recommended that research be continued in investigating human behavior of using the Sketch Model technique. It is recommended that an effort be carried out to identify meaningful naval command applications through which the Sketch Model technique can realize its promise as a decision aid.

EXPERIMENTAL INVESTIGATION OF SKETCH MODEL ACCURACY AND USEFULNESS IN A SIMULATED TACTICAL DECISION AIDING TASK

REPORT NO. 215-3

Contract No. N00014-75-C-0811

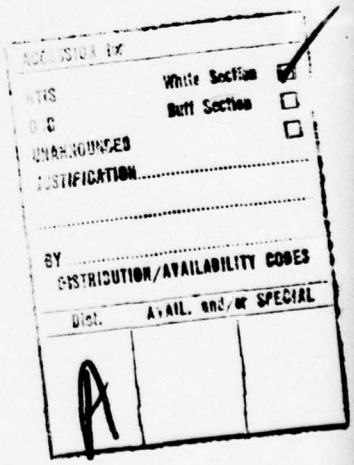
Prepared for:

Code 455
Director, Engineering Psychology Program
Psychological Sciences Division
Office of Naval Research
Department of the Navy
Arlington, Virginia 22217

By:

Gary W. Irving
Jennifer J. Horinek
Pak Y. Chan
David H. Walsh

Integrated Sciences Corporation
1640 Fifth Street
Santa Monica, California 90401



May 1977

EXECUTIVE BRIEF

A. BACKGROUND

This report describes the implementation and experimental investigation of a proposed aid to support naval tactical decision making. This decision aid, named the "Sketch Model," allows a human operator to communicate to a computer his subjective estimate of the form of any functional relationship that is continuous in at least one dimension. This communication is performed by the operator using an input device to electronically "sketch" the function on a computer graphics display.

The Sketch Model approach to subjective estimation is hypothesized to have certain advantages over comparable decision aiding techniques, such as the scalar-value representation of subjective judgments. These advantages include comprehensiveness of estimation; ease, speed, and accuracy of estimation and of updating; the ability to consider qualitative data; and the capability of allowing Bayesian estimation without explicit *a priori* data.

In a pilot study, operators were shown sets of points randomly sampled from known bivariate Gaussian probability density functions and asked to sketch (on the display) their estimates of the iso-probability contours of the parent distribution. The study indicated that humans can produce accurate (i.e., competitive with maximum likelihood estimation) Sketch Models of well-behaved continuous functions.

In the real world, however, functional relationships, though they may be "continuous in at least one dimension," are seldom "well-behaved," nor can they always be fully specified analytically. The study documented here was undertaken to extend the Sketch Model concept in two directions. First, the Sketch Model is applied to a tactical problem in which the function to be estimated is not well behaved. The function, instead, is not only multidimensional; it is also multimodal and unsymmetric. Second, the usefulness of decisions reached with the aid of Sketch Models is investigated.

B. PROBLEM SCENARIO AND IMPLEMENTATION

The problem used to study the usefulness of the Sketch Model as a decision aid was that of selecting the best flight path for an air strike against ONRODA Island, given (1) known locations and suspected types of enemy sensors, and (2) predetermined aircraft fuel allotments and speed versus fuel consumption characteristics. Enemy sensor performance was modeled in terms of detection rate as a function of distance from the sensor. A Sketch Model consisted of a set of the iso-detection rate contours of the composite detection rate surface produced by four sensors at given locations, but whose detection rate versus range capability may be imperfectly known.

The goodness of a strike path was measured by a utility function that reflected a trade-off between minimizing the probability of being detected along the path and maximizing the fuel remaining at the target. The primary functions for specifying the best strike path were modeling the composite detection rate surface produced by the enemy's sensors and optimizing the strike path with respect to the model. Four system concepts were defined, representing different allocations of these two functions: (1) modeling (without Sketch Model procedure) and optimization both allocated to the operator; (2) modeling (via the Sketch Model procedure) and optimization (aided by the Sketch Model) to the operator; (3) modeling (via Sketch Model procedure) to the operator and optimization (by a grid-oriented dynamic programming routine) to the computer; and (4) both modeling and optimization to the computer. This last allocation scheme provided what amounts to "answers" to the problem set.

C. RESULTS

Analyses of variance were performed on strike path utility data to compare system concepts. The results were weakened by the small number of subjects (4) available to provide the data. The conclusions evaluating the tactical usefulness of the decision aid were further weakened due to the problem set being too easy for the subjects. Within the qualifications posed by the small number of subjects and easy problem set, the result of the investigation of Sketch Model usefulness was that strike paths produced by computerized optimization operating on Sketch Models were significantly better than strike paths specified by subjects without the aid of Sketch Models.

The investigation of the accuracy with which humans could produce Sketch Models of messy functions was not affected by the easiness of the problem set. Sketch Model error was measured in terms of percent volume error from the true detection surfaces. Based on an examination of factors contributing to Sketch Model error, the findings relating to Sketch Model accuracy are:

1. Humans can use the present Sketch Model method to develop accurate models of multimodal, unsymmetric, three-dimensional functions.

2. Considerable improvement over present levels of accuracy can be achieved since the major sources of error can be reduced by refining the methodology.

3. Humans can use the Sketch Model method to significantly reduce the effects of uncertain information.

D. RECOMMENDATIONS

The primary recommendations regarding Sketch Model research with respect to the Operational Decision Aids program are listed below. In general, they reflect the desirability of more extensive experimentation and of further extensions of the Sketch Model concept.

1. Repeat the air strike path experiment, correcting for two weaknesses: number of subjects and easiness of the problem set.

2. Investigate means whereby the Sketch Model technique can be integrated with other classical decision theoretic techniques.

3. Extend the investigation of Sketch Models to examine the validity of the hypothesized advantages not yet addressed.

ACKNOWLEDGEMENTS

We would like to acknowledge the support of Dr. Martin A. Tolcott, Office of Naval Research. We especially appreciate the guidance and thoughtful suggestions offered by Dr. Tolcott and by Dr. H. Wallace Sinaiko, Smithsonian Institution.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

I. INTRODUCTION

The Operational Decision Aids (ODA) program, directed by the Office of Naval Research, is intended to develop and evaluate decision aids useful to the Task Force Commander and his staff. Integrated Sciences Corporation (ISC) is responsible for aspects of man-machine interface design, particularly (1) ways to increase the fluency of man-machine communications, and (2) ways to allocate functions between man and machine that take advantage of their respective strengths. ISC has developed a decision aid termed the "Sketch Model" that addresses these two factors. This report documents one of ISC's efforts to investigate the accuracy with which Sketch Models can be produced and their usefulness in a decision task.

A. BACKGROUND

In the operational environment, the decision maker must often deal with real-world relationships for which analytical models are either unavailable or inadequate. In the latter case, the available models (whether manual or computerized) may assume conditions that differ from those confronting the decision maker. In either case, by necessity or choice, the decision maker will tend to utilize his own perception of the relationship among the variables constituting the tactical problem. This perception can be thought of as a model, albeit an implicit one; like any other model, it can be used to aid the decision process. If this implicit model could be entered into the computer in a form that is both user-oriented and usable by the machine, it would provide input to a variety of analytical techniques best done by computer. The questions thus become how best to formalize (i.e., give definite form or shape to) a decision maker's implicit model and how best to transfer that now-explicit model into the computer.

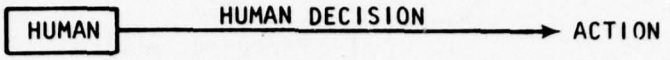
The Sketch Model concept was developed to address these questions. A Sketch Model provides a direct, straightforward technique for formalizing certain types of implicit models. Furthermore, when implemented on a computer-driven, interactive display, the Sketch Model is also readily available to the computer. The Sketch Model is essentially a "picture" that is

first mentally "visualized," then drawn by a decision maker. The picture represents the decision maker's perception of the functional relationship between two or more variables, with the stipulation that the function be continuous in at least one dimension. A Sketch Model can take a variety of forms; it must, however, be appropriate to the modeled relationship and easy for the decision maker to visualize and draw. Depending on the application, the Sketch Model can be, for instance, a single curve defining the relationship between two variables, or it can be a family of parameterized curves, or it can be a two-dimensional projection of the iso-"altitude" contours of a three dimensional function.

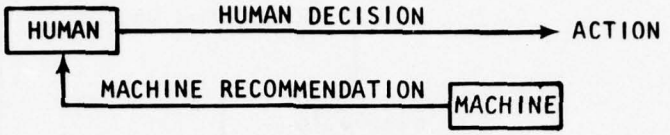
Of the three levels of machine participation in decision making shown in Figure 1, the Sketch Model technique belongs to the third one. The unaided, or "man-only," level depicted in the figure is relied upon when automated decision aids either do not exist or are inappropriate to the user's needs or preferences. Applications of the machine-aided, or "machine helping man," level have proliferated due to rapid advancements in computer capabilities and operations research techniques. Many of these applications, however, tend to be over-reliant on the machine. More recently, the "man helping machine to help man" type of decision aiding has evolved, allowing a better balance of man-machine task allocation according to their respective talents and limitations.

A decision task involves at least three basic functions: representation, evaluation, and optimization. Performing each of these functions can require model(s) of the real world. Depending on the design of the man-machine decision system, some of the required models will reside in the machine, where they may exist in the form of simulation processes, analytical models, or sets of rules; other required models can reside in the mind of the human decision maker. Figure 2 shows how these functions can be ordered in a feedback structure. Given a proposed solution to a tactical problem, the representation function is performed to predict an outcome on the basis of its constituent models. The evaluation function produces a measure of the preferability of that outcome. If the decision maker decides a better outcome is desired, the optimization function is performed to modify the originally proposed solution.

I. UNAIDED DECISION MAKING



II. MACHINE AIDED DECISION MAKING



III. HUMAN AIDED MACHINE DECISION RECOMMENDING

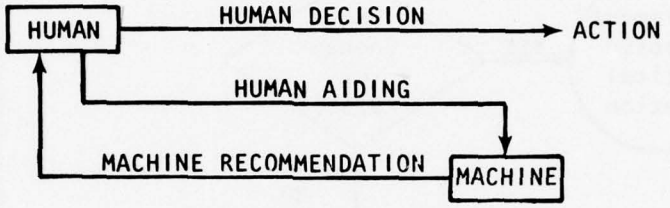


Figure 1. Levels of Aided Decision Making.

While the Sketch Model technique is potentially applicable to performing any of these three functions, its usefulness in carrying out the representation function is the most apparent. In a complex man-machine decision situation, the representation function will generally require multiple models, some of which may best be modified or supplied by some form of human subjective judgment. One familiar way to incorporate human subjective models into the representation function is to elicit subjective scalar values that stand for certain tactical relationships or factors. Scalar subjective judgment has been researched extensively and the conditions under which it can best be applied are fairly well known (References 1, 5). In a sense, the Sketch Model technique is an extension of the scalar subjective judgment approach. Both are instances of the "man helping machine to help man" approach to decision aiding; whereas a subjectively given scalar value provides a single evaluation of a factor (or the relationship between two factors) at a single point, the Sketch Model technique permits the value of

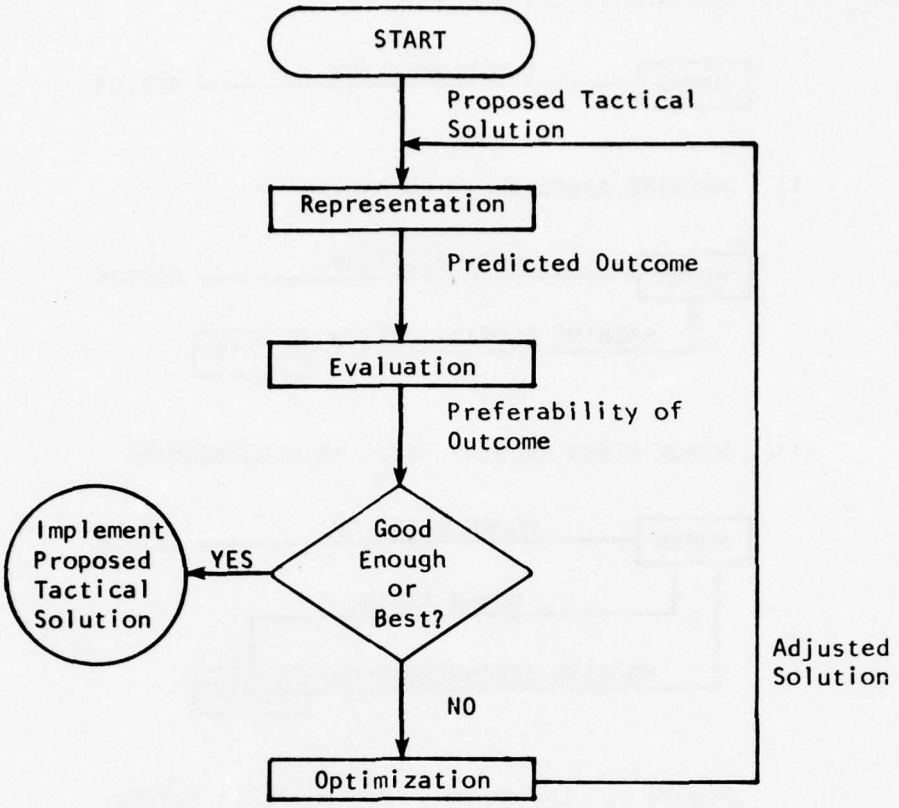


Figure 2. Feedback Structure of Basic Decision Making Functions.

a factor (or relationship among multiple factors) to be quickly determined at any allowable point. Clearly, it would require hundreds, perhaps thousands, of individually inputted scalar values to achieve the same richness of estimation as a single Sketch Modeled "picture" of a particular function. The Sketch Model technique is also potentially useful when available objective models are not applicable to the specific situation or when objective models do not exist. The hypothesized advantages offered by the Sketch Model technique are detailed in Table 1.

B. SUMMARY OF THE STUDY

Prior to this study, a preliminary investigation had evaluated the ability of human operators to generate Sketch Models of bivariate Gaussian density functions from sampled data. The results indicated that humans can

Table 1. Hypothesized Advantages of Sketch Model Technique Over Three Competitors.

I.	Compared to Theoretical Objective Modeling
	1. Updates more quickly and easily
	2. Integrates variables lacking comprehensive theory or difficult to quantify
II.	Compared to Empirical Objective Modeling
	1. Combines qualitative and quantitative data
	2. Allows nonlinear weighting of outliers
	3. Does not require parameteric functional expression
	4. Allows Bayesian estimation without explicit *a priori* data
	5. Counteracts biases in data sampling procedures
III.	Compared to Scalar Subjective Judgment Modeling
	1. Ease of estimation
	2. Speed of estimation
	3. Accuracy of estimation
	4. Ease of feedback

develop accurate Sketch Models of one type of well-behaved (i.e., unimodal and symmetric) three-dimensional function. Sketch Models developed from small samples of the underlying functions estimated those functions at least as well as, and in some cases better than, the empirical objective technique of maximum likelihood estimation. The current study was undertaken to extend those results by investigating the ability of human operators to generate Sketch Models of less well-behaved functions (i.e., multimodal and unsymmetric). This study also sought to evaluate the usefulness of those Sketch Models as aids to a tactical decision problem.

1. The Test Problem

The tactical decision problem was to come up with "best" air strike path through a defender's multi-sensor detection field. The goodness of each candidate strike path was computed according to a predetermined utility function; the "best" path was the one with the highest utility value. The

utility function was constructed to reward low probabilities of being detected along the strike path and high values of fuel remaining on target. The design of the experimental problems included these assumptions:

1. The location of each sensor is perfectly known.
2. Objective performance models (i.e., detection rate functions) exist for each type of single sensor.
3. Intelligence may err in identifying the type of sensor at any location.
4. Fuel consumption characteristics of the strike aircraft are known.
5. A function specifying composite sensor performance (i.e., detection rate value at any point between strike launch and target) cannot be arrived at by strictly analytical methods, due to such factors as intelligence errors.

This last assumption required that the decision maker furnish his subjective input as to the configuration of the composite detection rate field produced by the defender's sensors. Sketch Models were thus applied to specifying the composite detection rate surface from which the probability of being detected along the path could be computed and supplied to the utility function. The detection rate surface is three-dimensional, so that in this problem, the Sketch Model of the composite detection rate surface is a set of iso-"altitude" contours, each representing a percentage of the maximum detection rate value expected (also supplied by the decision maker). The appearance (and purpose) of the Sketch Model in this context is thus analogous to that of a contour map, where the altitudes can be thought of as detection rate values. The Sketch Models were generated on a computer-driven, interactive display via a control device (in this case, a trackball). Thus the display provided the operator with visual feedback on his model in the form of the detection contours he had drawn. He could also use these contours to help him devise strike paths, provided this task had been allocated to him. Furthermore, in this implementation the computer sampled the control device used to "draw" the Sketch Model, thus obtaining the data from which it could derive the cumulative probability of detection value needed by the utility function.

2. The Experiment

The two objectives of this study were to find out if human operators could produce accurate Sketch Models of a type of multidimensional, multimodal, unsymmetric function; and to ascertain the extent, if any, to which Sketch Models can aid the kind of decision task represented by the strike path optimization problem. A 48-problem set was defined, with each problem defined by sensor locations, intelligence estimate of sensor types, and an intelligence reliability level. Four system concepts were defined, representing alternative functional allocations and solution methods for devising strike paths. These were:

A. Operator specifies strike path without a Sketch Model of the composite detection field and without numerical evaluation feedback.

B. Operator first generates a Sketch Model of the detection field, then seeks the best strike path according to machine-computed utility of each path with respect to the Sketch Model.

C. Operator generates Sketch Model and machine uses a dynamic programming technique to solve for best path based on detection values computed from the Sketch Model.

D. Machine determines "true" detection contours of field and solves for "true" best path.

Data were collected from four subjects for each of the 48 problems under Concept B; the data consisted of "best" path utility for each problem and the subject's Sketch Model contours. Data were next collected from eight subjects under Concept A; data consisted only of subjects' best path utility for each problem. Next, using the four subjects' Sketch Models previously generated under Concept B, the machine-determined best path utilities were recorded (Concept C). Finally, the "true" detection contours and the "true" best path solutions were computed (Concept D).

Since four of the subjects participating under Concept A had been previously trained in Sketch Modeling and had been exposed to the problem set, two analyses of the path utility data were necessary. The first was a three-way analysis of variance on the path utility data generated under Concept A. The factors were groups--two groups of four subjects, one group trained in Sketch Modeling, the other naive; problem difficulty--four levels

corresponding to intelligence reliability levels; and replications--two levels. This analysis was required to determine whether data from Concepts A, B, and C could be validly compared; i.e., it was necessary to determine whether prior training in Sketch Modeling biased subjects' performance when Sketch Modeling was not a factor. The results indicated no biasing effect; in fact, the untrained group slightly outperformed the trained group.

The second analysis was a four-way analysis of variance performed to compare path utilities generated by the four experienced subjects under each of Concepts A, B, and C. Results showed that performance increased from Concept A to Concept C, but that only A and C showed a statistically significant difference. Further, when utilities from each method were compared to the "true" utilities gotten under Concept D, it was apparent that Concepts A, B, and C produced very similar performance, and that performance under any of those methods was in the vicinity of 90% of "perfect."

Two analyses were used to investigate Sketch Model accuracy. One was a three-way analysis of variance performed on the relative mean square errors between the Sketch Models and the "true" detection contours. This approach not only isolated an individual subject's performance, but also provided information on how well Sketch Models could adjust for increasing levels of uncertainty (represented in the problems by decreasing intelligence reliability). The second treatment relied on measurements of the percent volume error between the Sketch Models and the "true" detection function to provide insights into the absolute goodness of fit and to discern sources of Sketch Model error other than those evaluated by the ANOVA treatment. Results of the analysis of variance indicated that Sketch Model accuracy was affected only slightly by increasing levels of uncertainty. The goodness of fit data suggested that Sketch Model error is largely attributable to quantization error inherent in the present method and scalar subjective judgment error (from the peak detection rate estimation task). Consequently, the subjects' ability to generate accurate contours was not a substantial source of Sketch Model error.

3. The Findings

While the results provided some evidence that operators can generate fairly accurate Sketch Models of an irregular type of function, the experiment did not yield the kind of results that strongly suggest the usefulness (or lack thereof) of the Sketch Model technique. There were two reasons. One was insufficient data, due to a severe attrition in the pool of subjects trained in Sketch Modeling during a lengthy hiatus brought on by a series of hardware malfunctions. The second problem surfaced as soon as what data had been gathered were analyzed: it appears that the problems were too easy for the subjects. Despite considerable effort to ensure otherwise, it appears that the interaction between the utility function and the scenario elements produced problems in which the best air strike path was obvious. It is thus difficult to ascertain the usefulness of Sketch Models as an aid to the strike path decision problems. Although the best paths resulted from Concept C, which used Sketch Models, good paths also resulted from Concept A, which did not. There was also some indication that the Sketch Model technique in conjunction with computerized path optimization (Concept C) yields improved performance for subjects less astute at devising strike paths than at Sketch Modeling. This possibility, however, requires further substantiation.

Findings as to Sketch Model accuracy, on the other hand, were not affected by the triviality of strike path solutions and were generally positive. Based on data from the nearly two hundred Sketch Models generated, the findings relating to Sketch Model accuracy are:

1. Humans can use the present Sketch Model method to develop accurate models of multimodal, unsymmetric, three-dimensional functions.
2. Considerable improvement over present levels of accuracy can be achieved since the major sources of error can be reduced by refining the methodology.
3. Humans can use the Sketch Model method to reduce the effects of uncertain information.

C. THE REPORT

All phases of this study are documented in the following sections. Section II describes more fully the way the basic path optimization problem

was put together: it explains how the ONRODA Scenario was adapted, distinguishes at more length among the various ways of determining strike path solutions, and explains the analytical models for single sensor detection performance and aircraft fuel consumption, and characterizes the utility function developed to evaluate strike paths. Section III details system operation; it comprises step-by-step explanations of how path solutions were obtained by subjects, both with and without Sketch Models. Illustrations of the display formats and the appearance of Sketch Models used in this study are covered in Section III.B. Sections IV and V outline the experiment and the data analyses performed, respectively. Section VI interprets the results insofar as the data warrant. The appendices are used primarily to document the algorithms used to implement aspects of the study; they are provided for the reader interested in seeing how certain operations research techniques were adapted.

II. THE SKETCH MODEL APPLICATION

The tactical decision task defined to investigate the usefulness of Sketch Models as a decision aid was that of optimizing an air strike path through a defender's multi-sensor detection field. This section describes the task scenario, the four system concepts that represent different ways of optimizing a strike path, and the models of the scenario variables that constitute the experimental vehicle. Each model described reflects certain assumptions made about the behavior of the scenario variable. These assumptions, in turn, were adopted to keep the test vehicle simple, rather than to faithfully model the variables' "real world" performance. The utility criterion function, ultimately used as a performance measure, is also described here in terms of its supporting models.

A. AIR STRIKE SCENARIO

The problem selected, implicit in the ONRODA scenario, was that of optimizing an air strike path between a strike launch point and a target. The evaluation of the path depended on the probability of an aircraft's being detected by the enemy and the amount of fuel consumed by the aircraft along the strike path. Accordingly, certain elements of interest, particularly the scenario geography, were extracted from the ONRODA Warfare Scenario (Reference 4), and other details, described below, were added. The scenario developed here assumes that the decision has been made to conduct an air strike against ONRODA, so that investigating the Sketch Model approach in this study means applying it to one aspect of the operational implementation of the decision to strike.

Figure 3 shows the 500-by-500 n.m. portion of the ONRODA warfare scenario area map used to provide the geographical context for this study. The boundaries provide an area west of ONRODA for the selection of strike launch points and (it is assumed) enough room to plan strike paths that do not violate the ORANGE sanctuary.

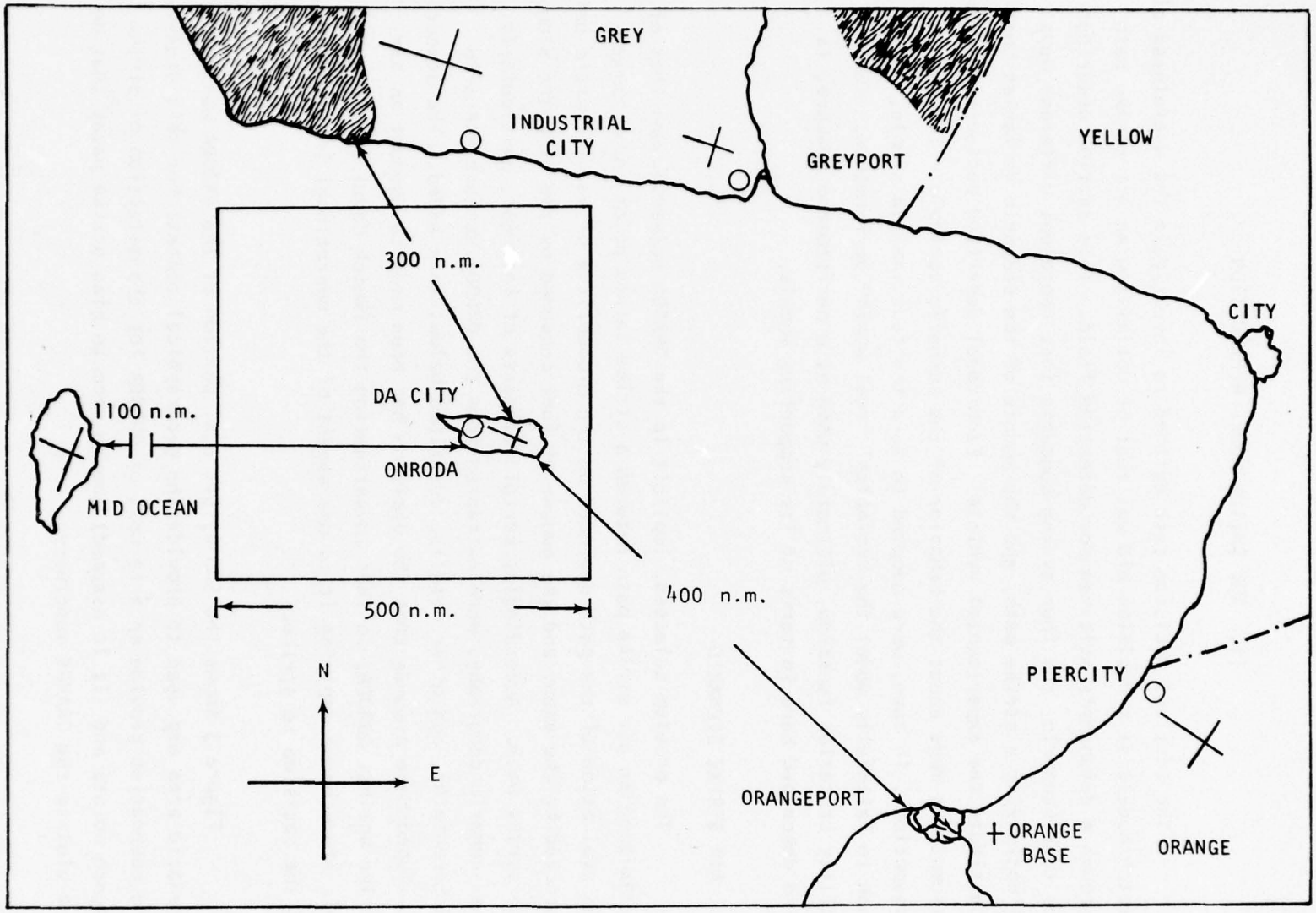


Figure 3. Strike Scenario Geography.

The air strike scenario used here incorporates some further assumptions. First, the strike target is taken to be the ONRODA airfield complex only. Second, the strike aircraft are supersonic, and they carry suitable stores and a predetermined fuel allotment.

The assumptions made about enemy defense have to do with the number, locations, and ranges of the ORANGE sensors that are capable of detecting the strike aircraft. Own intelligence reports that there are four such sensors and that their locations are pinpointed. One sensor is installed on ONRODA near the airfield. The other three are ocean-platform mounted, and, since ORANGE knows the general location of the task force, they are positioned west of ONRODA between the island and the task force. Intelligence reports that ORANGE has three types of sensors, with each type characterized by a different nominal detection range. However, unlike intelligence information on a sensor's location, which is accurate, own intelligence regarding a specific sensor's type (and hence its nominal range) may or may not be perfectly reliable (see Section IV.A.4).

The problem is to plan a strike path against the airfield on ONRODA that (1) minimizes the probability of a strike craft being detected, given the locations and types of enemy sensors, and (2) does not impose excessive fuel requirements on the aircraft, given its fuel allotment and its fuel consumption characteristics.

Further assumptions for this scenario are that neither the enemy's defense nor airborne enemy aircraft are to be considered explicitly as strike factors. Implicitly, enemy defense capability is one reason to minimize the probability of being detected along the strike path, tantamount to considering surprise as a strike factor. In a similar manner, attempting to postpone detection also affords less time for ORANGE aircraft on ONRODA or on the mainland to react, while attempting to conserve fuel enables the strike aircraft to maneuver if challenged by ORANGE aircraft after reaching the strike target.

B. DESCRIPTION OF SYSTEM CONCEPTS

Solving the strike path selection problem consists of two major functions: (1) modeling the composite detection field set up by the enemy's sensors, and (2) optimizing the strike path through that field. There are numerous ways to reach a solution, depending on how these two functions are allocated between man and machine and the methods used to perform each function once it is allocated.

For this study, four different allocation schemes were defined, each representing a man-machine system concept. Figure 4 categorizes these four system concepts according to their functional allocations. For the two system concepts (C and D) that allocate the optimization function to the machine, a dynamic programming algorithm was chosen as the appropriate optimization method. This choice was suggested by certain *a priori* assumptions about the nature of the solutions to the problem set, particularly that the detection surface might be multi-modal, a situation that dynamic programming methods could handle more easily than could other candidate methods.

The choice of dynamic programming to implement the machine-allocated optimization accounts for the empty cell at the top, right in Figure 4. Since System Concept A generates no explicit composite detection model, and since dynamic programming methods must optimize with respect to something (in this case, an explicit composite detection rate model), there was no way to implement a system concept according to the functional allocation represented by the empty cell's position. The allocation represented by the second empty cell, i.e., modeling performed by machine and optimization performed by a human operator, was not considered, partly to keep the study manageable, and partly because the study focused on the operator's modeling performance and how that affected optimization performance rather than on optimization techniques themselves.

The four system concepts defined by the functional allocations in Figure 4 are described below:

1. System Concept A. Both the detection field modeling and strike path optimization functions are allocated to the human operator. In this case, the operator's model is not made explicit, i.e., there is no

MODELING ALLOCATED TO:	OPTIMIZATION ALLOCATED TO:	
	OPERATOR	MACHINE (via Dynamic Programming)
OPERATOR[1]	SYSTEM CONCEPT A	
OPERATOR[2]	SYSTEM CONCEPT B	SYSTEM CONCEPT C
MACHINE		SYSTEM CONCEPT D

[1]Model implicit (i.e., no composite detection contours)

[2]Model explicit (i.e., Sketch Model applied in the form of composite detection rate contours)

Figure 4. System Concepts in Terms of Functional Allocations.

sketch-inputted representation of the operator's mental construct of the *composite detection rate surface. Since there is no explicit modeling* and since no performance feedback is provided, this system concept is referred to as "operator-unaided," and is intended to represent (in the context of this study) the way a strike path selection problem might be currently approached in the operational environment.

2. System Concept B. This concept also allocates both the modeling and optimization functions to the operator. In this case, however, the operator represents his model explicitly through the Sketch Model input procedure (Section III.B) and attempts to optimize a strike path with respect to that model. Performance feedback, in the form of a path utility, is provided to the operator for each strike path alternative he explores by exercising the path against the Sketch Model.

3. System Concept C. Under this concept, the modeling function is allocated to the operator, and the optimization is allocated to the computer,

where it is performed by a dynamic programming routine. The operator provides the computer with his model, which he makes explicit through the same Sketch Model input procedure used in System Concept B; the machine-assigned optimization is performed on that model.

4. System Concept D. This concept allocates both the modeling and optimization functions to the computer. Using the single sensor detection rate models (Section II.C.1) and perfect information as to location and type of each sensor, the computer determines a "true" set of composite detection contours. The dynamic programming routine then optimizes a strike path with respect to this "true" model. In the context of this study, Concept D provides what amounts to "answers" for both the modeling and optimization portions of the problem. The "answers" generated by this system concept are simply benchmarks against which to compare aspects of performance under the other three system concepts; such "answers," of course, are unobtainable in the real world.

Comparing the results of each of these concepts when applied to the strike path problem provides insights into the effectiveness of the Sketch Model technique. Since the functional allocation represented by Concept A, for instance, resembles that of current operational procedures, the results of applying Concept A to the strike path problem can indicate what, if any, performance increment over present procedures can be achieved by using Sketch Models to solve the same problem. The inclusion of System Concept C serves to isolate the operator's performance on the modeling function from his performance on the optimizing function conducted under B. The results of applying System Concept D provide information on the absolute effectiveness of both the modeling and optimization functions carried out under the other three system concepts. In addition, the results of C and D considered together are intended to indicate which, if either, of the two functions performed by the operator under B is more appropriate to man or computer.

C. SIMULATION MODELS

The "goodness" of a path generated under any of the four system concepts depends on two factors: fuel consumed along the path and the cumulative probability of being detected. In order to compute a numerical value (or utility) that reflects a given path's "goodness," it is first necessary to have some way of quantifying those two factors. This was provided by a set of simulation models and computational algorithms developed for the study. Fuel consumption was modeled as a single functional relationship. The cumulative probability of being detected, however, is more complex and depends on how the characteristics of the detection field are defined. In general, this involves first defining single sensor performance, then defining the way a number of these single sensors combine to create a composite detection field. In other words, modeling the composite detection field either with or without Sketch Models requires other, more basic, underlying models.

The set of models developed for the study includes:

1. Single Sensor Detection Rate Model
2. Cumulative Probability of Being Detected Algorithm
3. Fuel Consumption Model
4. Utility Criterion Function
5. Dynamic Programming Algorithm
6. "Where-Am-I" Algorithm
7. True Detection Rate Contour-Drawing Algorithm

Numbers 1 - 5 were basic to the implementation of each of the system concepts and are described in this section. The use of numbers 5 - 7 depended on which system concept was being exercised; they are documented in the appendices.

1. Overview

Figure 5 depicts an example of how the models are interrelated. The figure is based on System Concept B since that concept uses all the models except for the dynamic programming algorithm. (Figures for the other concepts would be similar.) In general, Figure 5 shows that the scenario elements (sensor locations, reported types, strike launch point, and intelligence reliability) defining the problem are shown to the operator ((1)). The

operator then uses the Single Sensor Detection Rate model ((2)) to formulate his implicit model of the composite detection field. He then enters his Sketch Model of it via the display ((3)). Next the operator generates his proposed path ((4)). Values from the Sketch Model and the path are processed by the "Where-Am-I" algorithm and the P_d model to provide the utility function with the cumulative detection value it needs ((5)). In addition, path data is used by the fuel consumption model, which then provides to the utility function a value for the total fuel used ((5)). The utility function combines these two values and returns a single numerical value for the path's "goodness" ((6)). The operator reiterates (4) (i.e., draws another path which is evaluated by (5) and (6)) until he is satisfied that he has defined the best possible path.

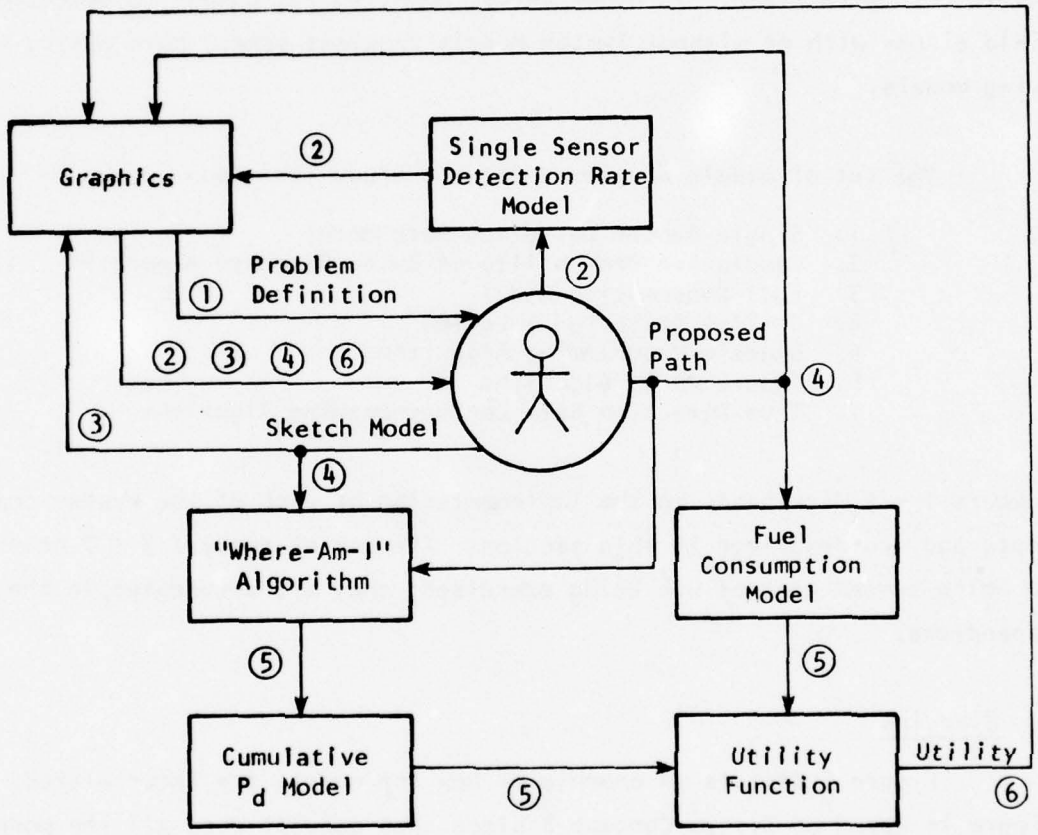


Figure 5. Example of the Interrelation of Models and Algorithms (for System Concept B).

2. Detection Rate Model

The detection capability for a single human-operated sensor is modeled as a detection rate,* which gives the probability of detection per time unit. The detection rate is assumed to vary as a function of range from the sensor. This relationship can be quantified according to:

$$\gamma(R) = \frac{\gamma_{max}\sqrt{e}}{R_{max}} R \cdot \exp\left(\frac{-R^2}{2R_{max}^2}\right) \qquad (1)$$

where

$\gamma(R)$ = the value of detection rate at radial distance R from sensor

γ_{max} = maximum detection rate for the sensor

R_{max} = range from sensor at which γ_{max} occurs

The general shape of detection rate-versus-range curve as governed by Equation (1) is shown in Figure 6. Equation (1) for $\gamma(R)$ models a sensor with a maximum detection rate γ_{max} at range R_{max} from the sensor. From the sensor location (where the detection rate is zero) detection rate increases monotonically until it reaches its peak value at range R_{max}. From this range, R_{max}, the detection rate drops off monotonically moving away from sensor until it reaches zero at some range beyond R_{max}. Hence, if we visualize $\gamma(R)$ as a three-dimensional surface it would look like a volcano with a hole at the center, where the sensor is located. Around this hole is a circular ridge at a radial distance R_{max} from the center of the hole. Beyond the ridge the sides of the "volcano" slope downwards until "ground level" is reached.

For the experiment three types of sensors were defined, corresponding to R_{max} = 37.5, 50, and 62.5 nautical miles and called small, medium, and large sensors, respectively. (Note that these values of R_{max} were selected for their suitability to the study, and that they were not intended to be the performance values for any "real world" sensor.) The maximum detection

*For Δt small enough, we can assume the detection rate γ to be constant within Δt. Then $\gamma\Delta t$ is the probability of detection during time interval Δt. For appropriate choice of Δt, $\gamma\Delta t \leq 1$.

rate γ_{max}, was 0.1 for each sensor type.* The performance curve for each sensor type is shown in Figure 7.

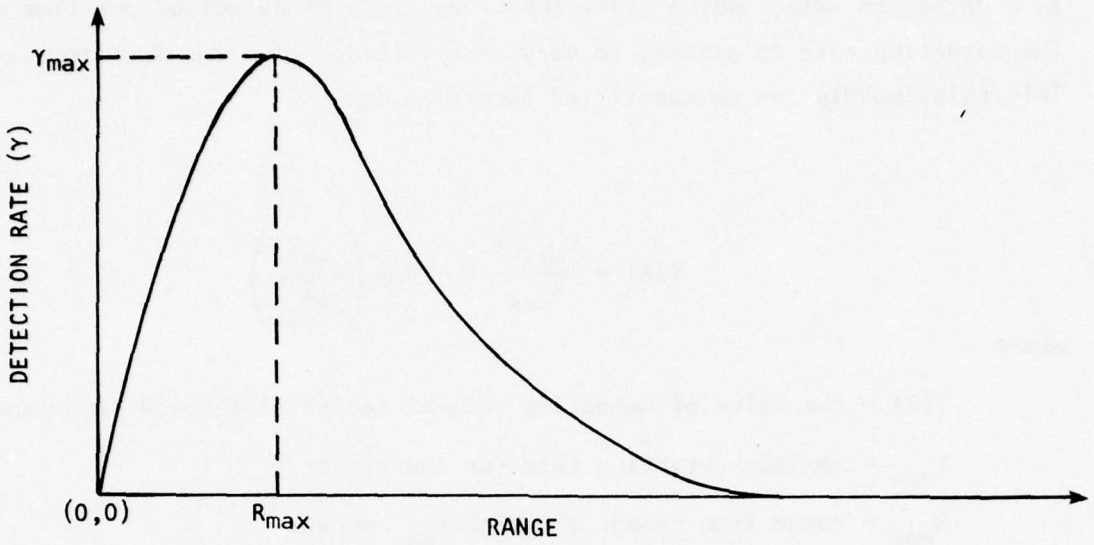


Figure 6. Single Sensor Detection Rate as a Function of Range.

Recall that the scenario specified four enemy sensors deployed, so that if two (or more) detection ranges overlap, we are really concerned about our strike craft being detected by either rather than being detected by both. In other words, we are concerned about the total detection rate at any point that any given combination of sensor locations and types will produce. This composite detection rate is easily computed. The composite detection rate γ_c at a point (x,y) is the sum of detection rates at (x,y) due to each sensor. Hence, if $\gamma_i(x,y)$ is the detection rate due to sensor i, the composite detection rate at (x,y) is

$$\gamma_c(x,y) = \sum_i \gamma_i(x,y) \qquad (2)$$

Each term $\gamma_i(x,y)$ on the right hand side of Eq. (2) is obtained by transforming the radial coordinates of Eq. (1) into rectangular coordinates.

*In the experiment the subjects were told to assume γ_{max} to be 0.2 because during a series of previous trainings they were accustomed to this particular value, and it would be less confusing if the value of 0.2 continued to be used. The computer software for the experiment then scaled γ_{max} to 0.1 in a manner that did not affect the performance of the subjects.

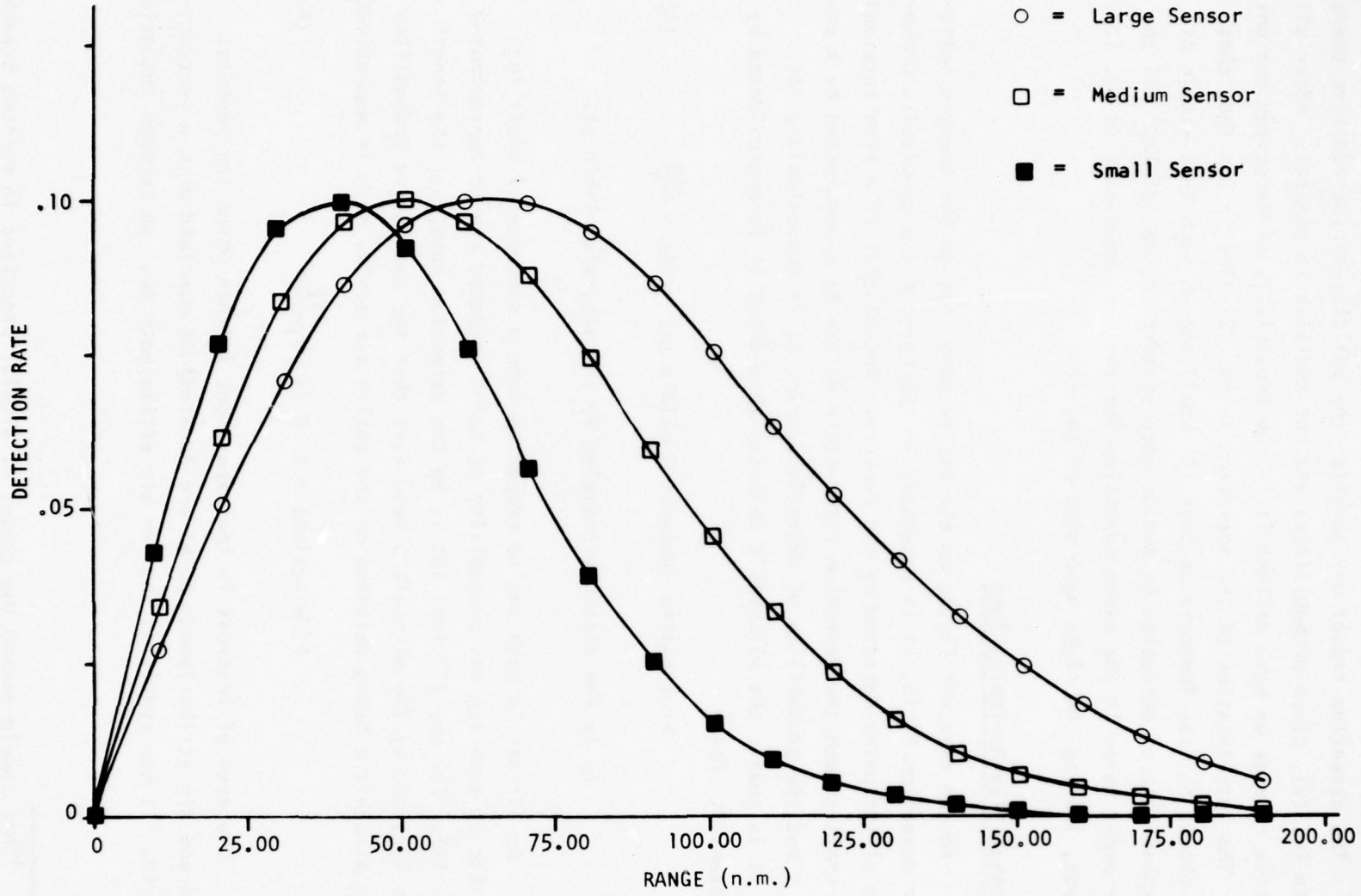


Figure 7. Performance Curves of Modeled Sensors.

The attentive reader may question the validity of the additive operation in Eq. (2), since probabilities are not additive in general. After all, detection rate as we have defined it is the probability of detection per unit time. The justification of the operation in Eq. (2) lies in the fact that if we choose Δt (see footnote on page 19) small enough such that within Δt the probability of detection by two or more sensors is negligible,* all the higher order terms in the exact expansion for the left hand side of Eq. (2) drop out, leaving the right hand side of Eq. (2).

3. Probability Approximations

For an aircraft flying an air strike path through the enemy's multi-sensor detection field, it is necessary to approximate the cumulative probability of its being detected by the sensors. Recall that if a time interval Δt is small enough the detection rate within Δt can be approximated by a constant γ and the probability of detection during Δt is approximately $\gamma\Delta t$. Since Δt is small, the aircraft's velocity throughout Δt is approximated by a constant v. Hence

$$\text{Probability (detection within } \Delta t) \simeq \gamma\Delta t \simeq \gamma\frac{\Delta s}{v} \tag{3}$$

where

Δs is the distance traveled by the aircraft within Δt.

An air strike path can be broken up into a sequence of small legs such that on each leg the probability of being detected can be approximated by Eq. (3). For the i^{th} leg, let γ_i be the detection rate, Δs_i the length of the leg, and v_i the aircraft's velocity; then the cumulative probability of the aircraft's being detected on the entire air strike path is approximately

$$P(\text{detection}) \simeq 1 - \prod_i \left(1 - \gamma_i\frac{\Delta s_i}{v_i}\right) \tag{4}$$

The area of interest in the experiment (within which the sensors, target and air strike launch point are located) is overlaid with a rectangular grid. It was specified that an air strike path must go through the grid

*This should remind the reader of similar practices in various branches of operations Research, such as queueing theory.

points (see Section III), hence the grid points actually serve as "way points" that define a path as a series of path legs. By selecting an appropriate grid size, the approximations in Eq. (3), hence Eq. (4), hold.

There are two basic approaches to determining the detection rate γ_i along each path leg in Eq. (4). The first is to sample the detection rate at the midpoint of a leg, and use it as the γ_i for the i^{th} leg. The second is to sample the detection rates at the two end points of a leg, and use the average of these two values as the γ_i. When operating with a true detection rate field, where Eq. (2) in Section II.C.2 is used to calculate detection rate, we used the first method to sample γ_i. When operating on a Sketch-Modeled detection rate field we chose the second sampling method for γ_i. This is because a Sketch Model is discrete in the detection rate dimension so that using the first method would distort the results. In order to determine the detection rate at any point on a Sketch-Modeled detection surface, the "Where-Am-I" algorithm has to be exercised (See Appendix A).

4. Fuel Consumption Model

The fuel consumption model used was discrete. Three allowable velocities and their corresponding fuel consumption rates were selected. They are listed in Table 2. (Note that these values were selected only to correspond to the assumption that the strike craft are supersonic; they are not intended as performance data of any actual aircraft.) The second and third columns of Table 2 are equivalent; they are simply expressed in different units for easier reference. Hence, an aircraft traveling at a speed of 715 knots consumes 2.93 lbs. of fuel per second, or 14.8 lbs. of fuel for each nautical mile covered. In the experiment the three velocities were labeled as low, medium, and high, and these were the only allowable velocities.

Table 2. Fuel Consumption Rates.

Velocity (knots)	Fuel Consumption Rates	
	lbs/sec	lbs/n.m.
195	.574	10.6
715	2.93	14.8
1430	17.60	44.3

The amount of fuel that the aircraft carries on each mission is proportional to the range from the strike launch point to the target. Thus, if the range is doubled, the fuel allowance for the mission also doubles. Forty pounds of fuel were allowed for each nautical mile between the air strike start position and target. This permits the aircraft to do some high-speed maneuvering, but sustained high-speed travel is discouraged by the fact that allotted fuel would run out before the aircraft could accomplish the mission or return to the carrier.

5. Utility Criterion Function

A utility criterion function with which to measure the performance under each of the system concepts in the experiment was defined. The problem was to select an optimal air strike path, so an appropriate utility criterion function is one which measures the "goodness" of such an air strike path. The two variables selected to determine the goodness of an air strike path were fuel consumption along the path and probability of being detected by the enemy sensors (Sections 3 and 4, preceding). Since the utility function was prespecified to measure the goodness of any proposed path, no inputs were elicited from operators as to desirable values of the two variables. The following definition of the utility criterion function, U, incorporates a tradeoff between minimizing the probability of being detected by enemy sensors on one hand and maximizing the fuel remaining upon arrival at the target on the other.

$$U(F,P) = \left[\frac{(a-b)D - F}{2(a-2b)D}\right]^{(.01 + 2.475P)}, \text{ if } (a-b)D - F \geq 0 \qquad (5)$$

$$= 0 \text{ if } (a-b)D - F < 0$$

where

F = total amount of fuel consumed upon arrival at target

P = cumulative probability of being detected by enemy sensors

D = distance between strike launch point and target

a = fuel allowance/n.m.

b = fuel consumption/n.m. at an achievable speed resulting in the lowest fuel consumption per unit distance traveled

For each mission the fuel allowance is proportional to the shortest distance between the air strike launch point and the target ($a \cdot D$). The absolute minimum fuel that has to be preserved in order to return from the target is ($b \cdot D$). Hence, $(a - b)D$ is the maximum amount of fuel available for maneuvering to the target, and $(a - 2b)D$ is the maximum amount of fuel remaining upon return to the carrier. Note that if the aircraft runs out of fuel before returning to the carrier, the resulting utility is zero. For the experiment, $a = 40$ lbs/n.m., and $b = 10.6$ lbs/n.m., corresponding to a velocity of 195 n.m./hr.

The utility function takes on any value between 0 and 1, with higher utility values corresponding to "better" paths. A family of parameterized curves from the utility function is shown in Figure 8. The figure shows that as the probability of being detected by enemy sensors decreases, the utility value goes up. Also, if the probability remains constant, the utility value increases as fuel consumption drops. It is obvious why it is desirable to minimize the probability of being detected by enemy sensors. The rationale for encouraging fuel preservation is that if detection occurs at any time up to arrival at the target, there should be as much fuel left as possible in order to do some flight maneuvering to try to return safely.

In general the two goals of minimizing fuel consumption and minimizing the probability of being detected are incompatible. A nontrivial optimal air strike path thus requires a reasonable compromise between the two goals. The utility function was designed as representative of the class of functions useful in designing an air strike path through a multi-sensor field, and Eq. (5) embodies trade-off between remaining fuel and cumulative probability of being detected.

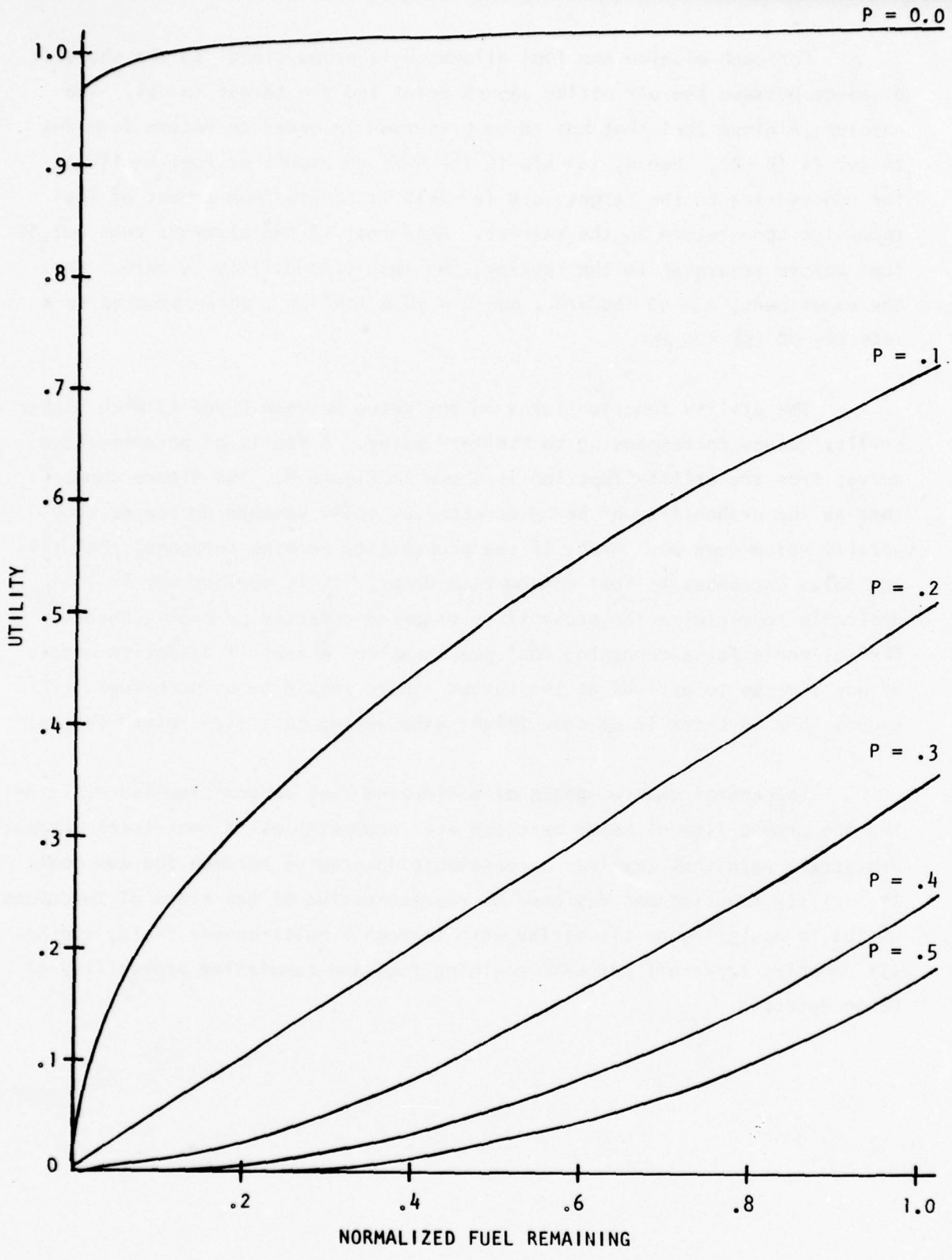


Figure 8. Family of Parameterized Utility Function Curves.

III. SYSTEM OPERATION

System Concepts A and B, both of which allocated the modeling and optimization functions to the human operator, were implemented using a computer-driven, interactive, color display. This section documents the step-by-step operation of both system concepts and describes certain features of their respective user interfaces, with emphasis on the display formats used.

The descriptions overlap at certain points. As implemented, both system concepts used the same hardware elements. These were, in addition to the display, a light pen, a trackball, a function keyboard, and (under B) an alphanumeric keyboard. On the basis of a prior study (Reference 2), the trackball served as the primary input device. As will be seen below the trackball was used to input path solutions and, under System Concept B, to input the operator's Sketch Model. The light pen and alphanumeric keyboard were used for auxiliary control and input tasks; the function keyboard was used to control both the flow of system operation and certain of the display format features. Display formats, including the use of color, were also similar at certain stages of the operation of each system concept. The discussion that follows includes illustrations of the display format at each stage of each system operation.

A. SYSTEM CONCEPT A

System Concept A called for the operator to (1) form an <u>implicit</u> estimate of the composite detection rate field of a given combination of sensor locations and reported (though not necessarily reliably) sizes; and then (2) specify an air strike path that he judged optimal with reference to his implicit composite detection rate model. A display format feature, in the form of "sensor coverage templates," was provided to support the modeling function. Except in the training phase (Section IV.B), however, performance feedback features were not incorporated in the operation of this system concept.

For convenience, the discussion of the operation of System Concept A to solve a single problem is divided into two segments: sensor coverage investigation, and strike path specification.

1. Sensor Coverage Investigation

Although System Concept A did not call for the operator to make explicit his model of the composite detection rate surface, it did allow him to represent on the display screen his estimate of the sensor coverage configuration. "Sensor coverage configuration" under this system concept refers to any combination of four "sensor templates," one centered over each sensor location. A "sensor template," in turn, refers to a circle that shows the distance from the sensor location at which the sensor's detection rate is 10% of its maximum (refer to Section II.C.2). This portion of the system operation description explains the procedure followed to arrive at an estimate of the sensor coverage configuration and illustrates the results.

a. Display Format. At the beginning of each trial, the display presents the information as shown in Figure 9. The situation geography (the screen area within the frame boundary) includes the location on ONRODA Island of DA City, an airstrip on the island (coded as a red, slanted cross), and one sensor installed on the island (coded as a green cross). In the "ocean" area below ONRODA Island appear the locations of three additional ocean-platform-mounted sensors, also coded as green crosses. Associated with each sensor location is a letter code designating the intelligence estimate of that sensor's type in terms of detection range: S corresponds to small, M to medium, and L to large.

Two types of information are displayed outside the frame boundary. First, the reliability of the intelligence information about the sensor coverage sizes is given. Recall from Section II.A that intelligence may not be perfectly reliable. There were four qualitative descriptors characterizing various levels of confidence in the intelligence information: perfect, very confident, confident, and not confident. The intelligence reliability estimate given for the problem illustrated in Figure 9, for instance, is given as "confident." To the operator, this means that the letter code for coverage size that appears with each sensor has some probability of being incorrect. Although the operators were not briefed on the actual details of the intelligence reliability model (described in Section IV.A.4), they did receive feedback on "true" sensor coverage sizes during their training (Section IV.B.1) from which they were encouraged to infer

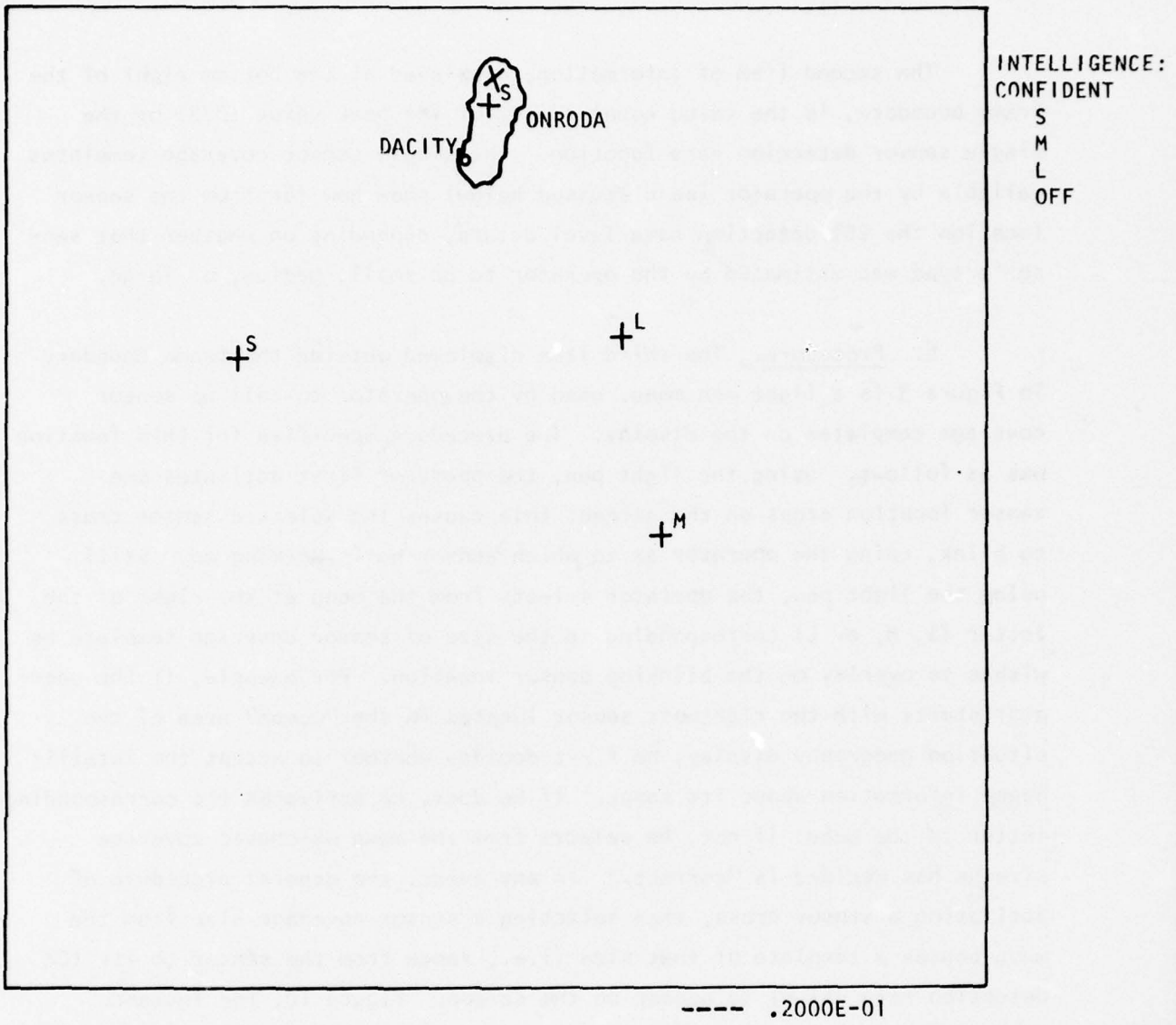


Figure 9. Display at Problem Initialization, System Concept A.

patterns of intelligence reliability. (The way in which the intelligence reliability assessment might be used during system operation is described later.)

• The second item of information, displayed at the bottom right of the frame boundary, is the value equal to 10% of the peak value (0.2) of the single sensor detection rate function. The single sensor coverage templates callable by the operator (as discussed below) show how far from the sensor location the 10% detection rate level occurs, depending on whether that sensor's type was estimated by the operator to be small, medium, or large.

b. Procedure. The third item displayed outside the frame boundary in Figure 9 is a light pen menu, used by the operator to call up sensor coverage templates on the display. The procedure specified for this function was as follows. Using the light pen, the operator first activates one sensor location cross on the screen; this causes the selected sensor cross to blink, cuing the operator as to which sensor he is working on. Still using the light pen, the operator selects from the menu at the right of the letter (S, M, or L) corresponding to the size of sensor coverage template he wishes to overlay on the blinking sensor location. For example, if the operator starts with the rightmost sensor located in the "ocean" area of the situation geography display, he first decides whether to accept the intelligence information about its range. If he does, he activates the corresponding letter in the menu; if not, he selects from the menu whichever coverage size he has decided is "correct." In any event, the general procedure of activating a sensor cross, then selecting a sensor coverage size from the menu causes a template of that size (i.e., range from the sensor to its 10% detection rate value) to appear on the screen. Figure 10, for instance, depicts the screen appearance after the operator has called up a template for the rightmost ocean-located sensor. Note that in this example he has disregarded the intelligence estimate of that sensor's coverage (given as medium), and has instead called up a large template for that sensor. Templates are called up in turn until all four sensor locations are overlaid. If the operator decides to change any of them he can do so by activating the selected sensor cross and then either selecting one of the other two sizes from the menu or using the "Off" option on the menu to erase a template, then making

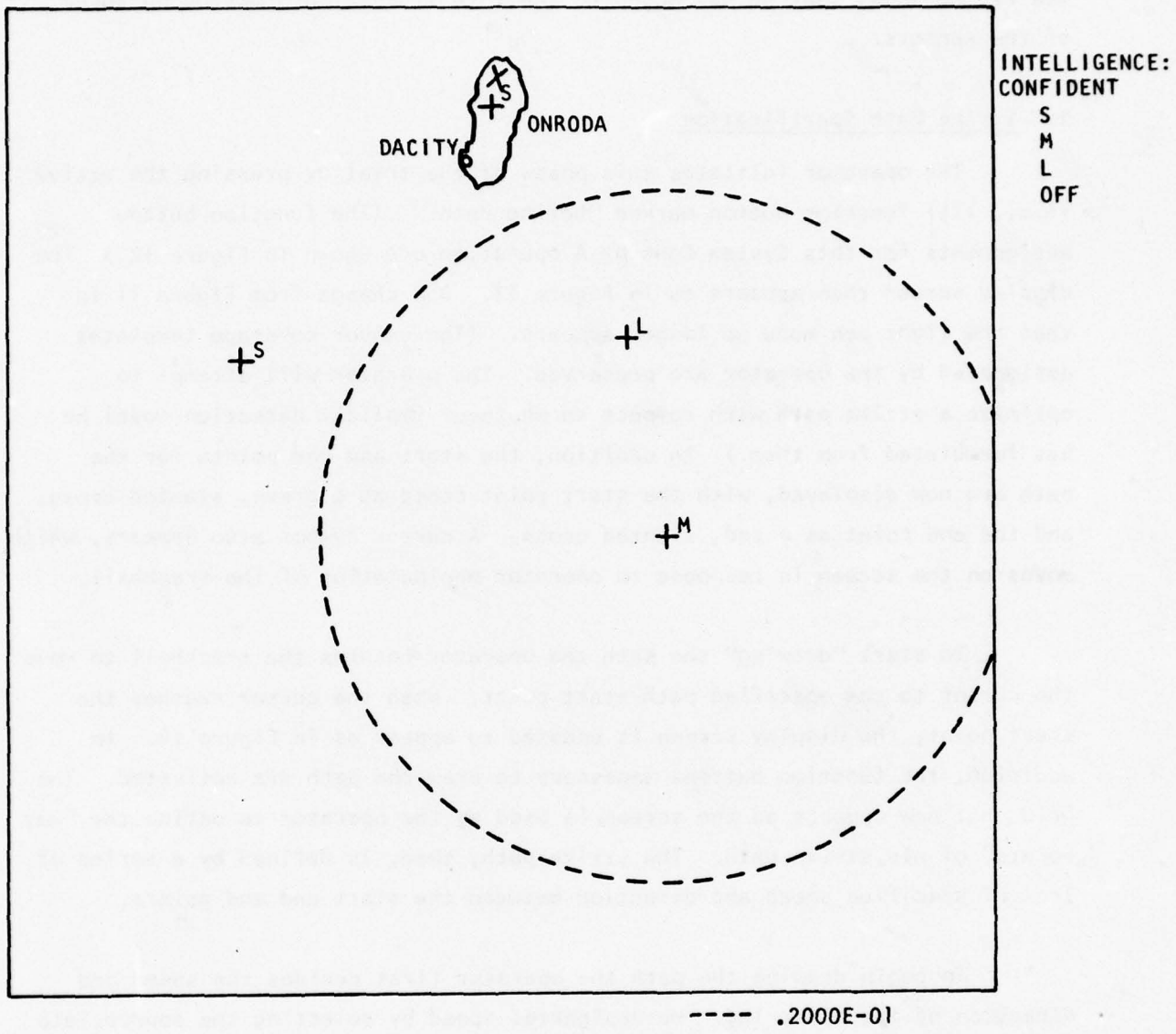


Figure 10. Display with One Sensor Coverage Template Selected, System Concept A.

another choice of template size. Figure 11 shows the screen with the sensor coverage template configuration as the operator might be satisfied with it. The figure shows that he has accepted the intelligence information on three of the sensors.

2. Strike Path Specification

The operator initiates this phase of the trial by pressing the active (i.e., lit) function button marked "Define Path." (The function button assignments for this System Concept A operation are shown in Figure 12.) The display screen then appears as in Figure 13. One change from Figure 11 is that the light pen menu no longer appears. (The sensor coverage templates designated by the operator are preserved. The operator will attempt to optimize a strike path with respect to whatever implicit detection model he has formulated from them.) In addition, the start and end points for the path are now displayed, with the start point coded as a green, slanted cross, and the end point as a red, slanted cross. A cursor symbol also appears, which moves on the screen in response to operator manipulation of the trackball.

To start "drawing" the path the operator rotates the trackball to move the cursor to the specified path start point. When the cursor reaches the start point, the display screen is updated to appear as in Figure 14. In addition, the function buttons necessary to draw the path are activated. The grid that now appears on the screen is used by the operator to define the "way points" of his strike path. The strike path, then, is defined by a series of legs of specified speed and direction between the start and end points.

To begin drawing the path the operator first decides the speed and direction of the first leg. He designates speed by selecting the appropriate function button, marked L, M, or H for low, medium, or high speed. When a speed has been selected, the legend "Select Speed" (shown in Figure 14) disappears and the legend "Draw Path" flashes at the lower left edge of the screen. This cue indicates that the cursor can now be moved via trackball manipulation to any eligible way point on the grid. Any one grid point adjacent to the current cursor position is eligible for selection by the operator; any non-adjacent point is ineligible and the cursor is prevented by software from going between or past currently eligible points. Figure 15 shows the

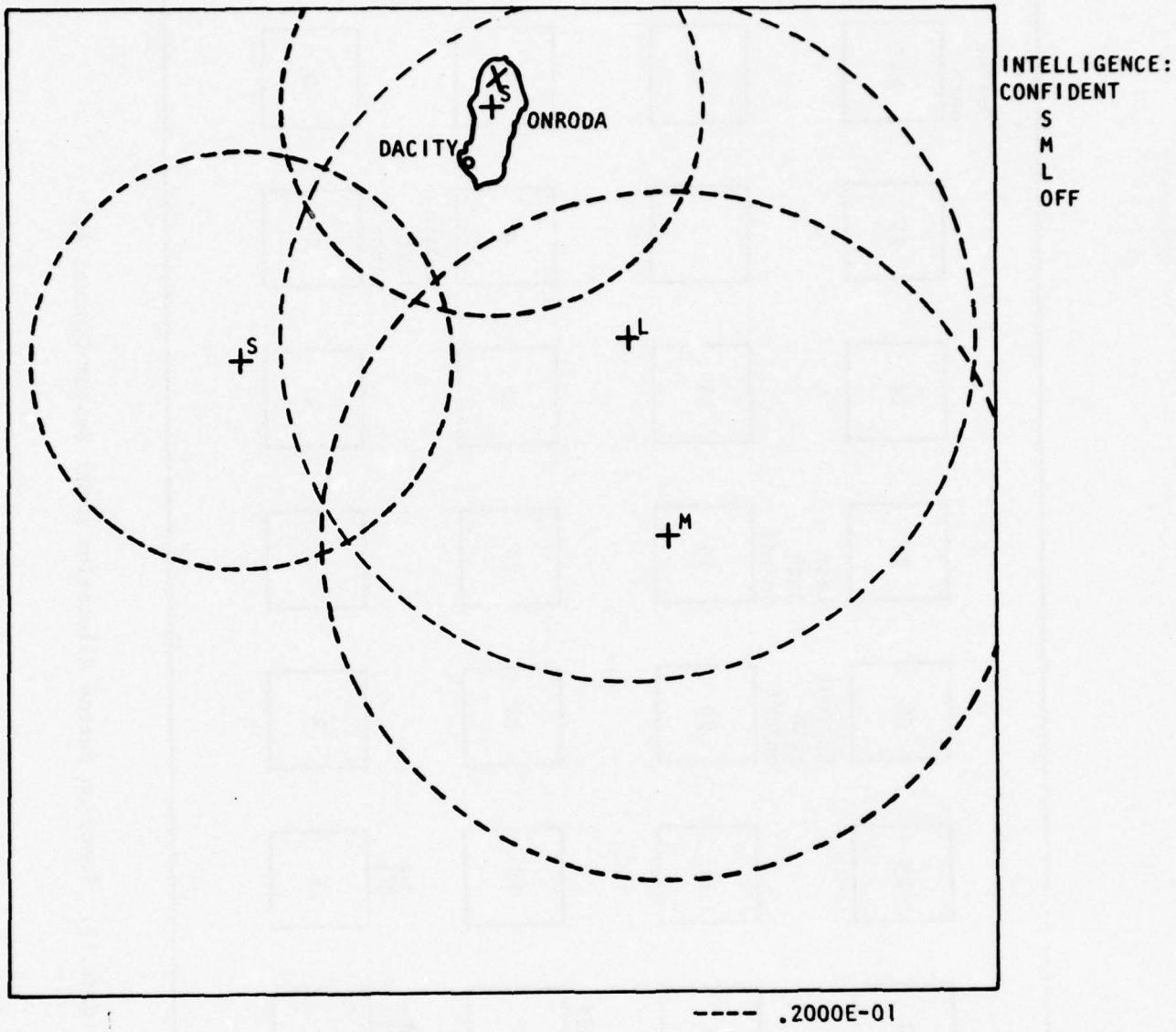


Figure 11. Display with All Sensor Coverage Templates Selected, System Concept A.

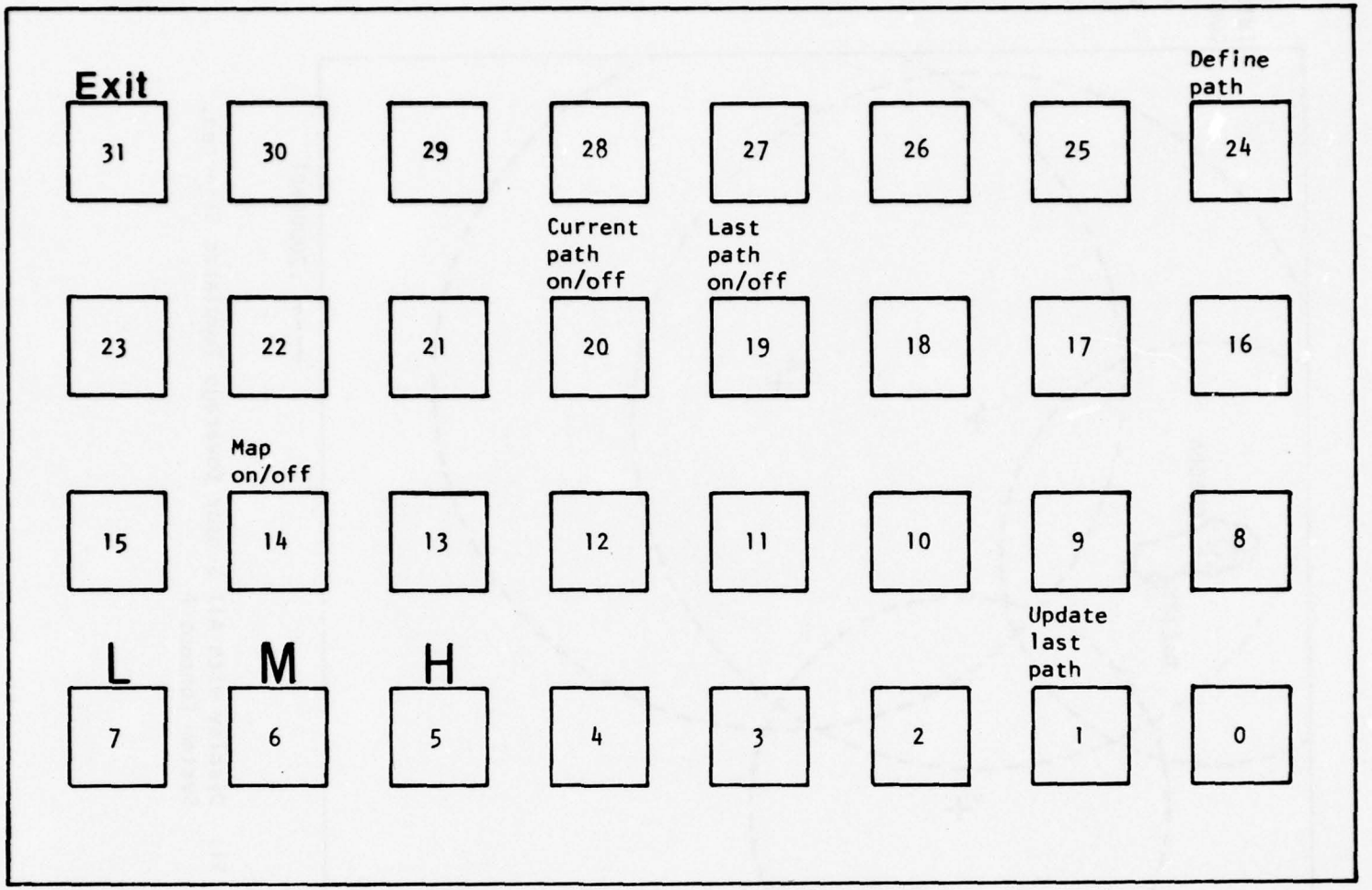


Figure 12. Function Button Allocation under System Concept A.

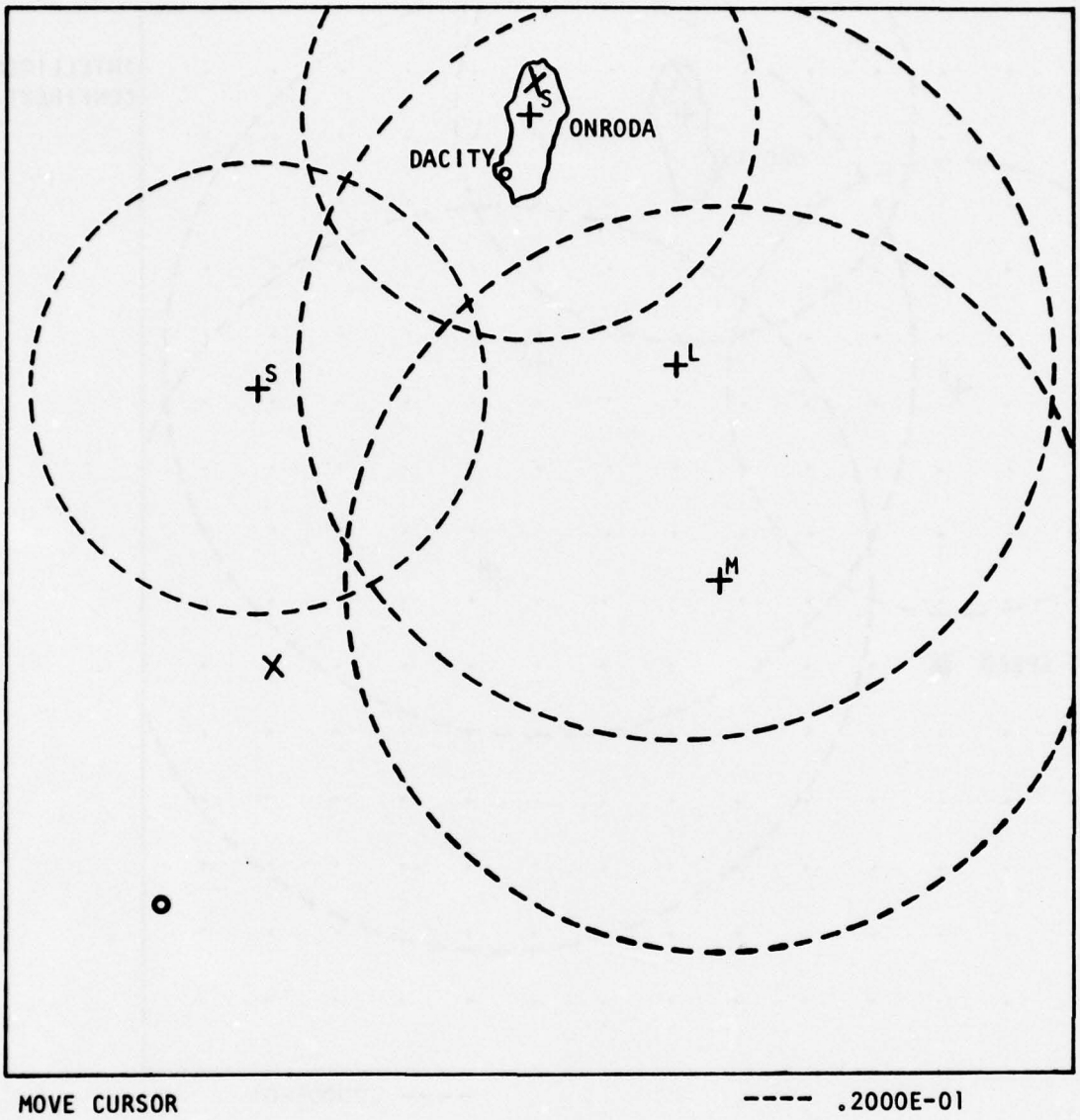


Figure 13. Display at Start of Strike Path Specification Phase, System Concept A.

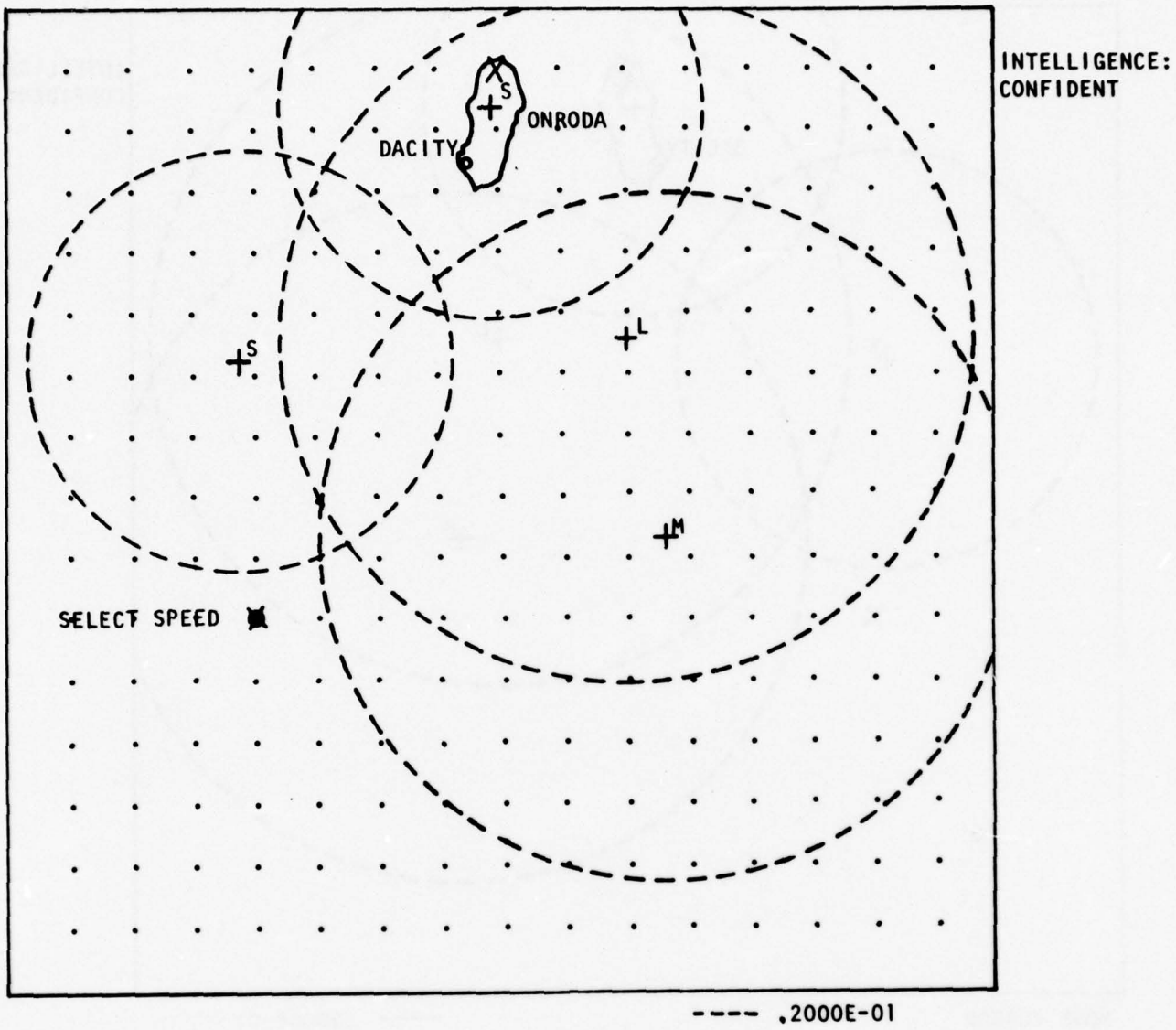


Figure 14. Display with Path Way Point Grid, System Concept A.

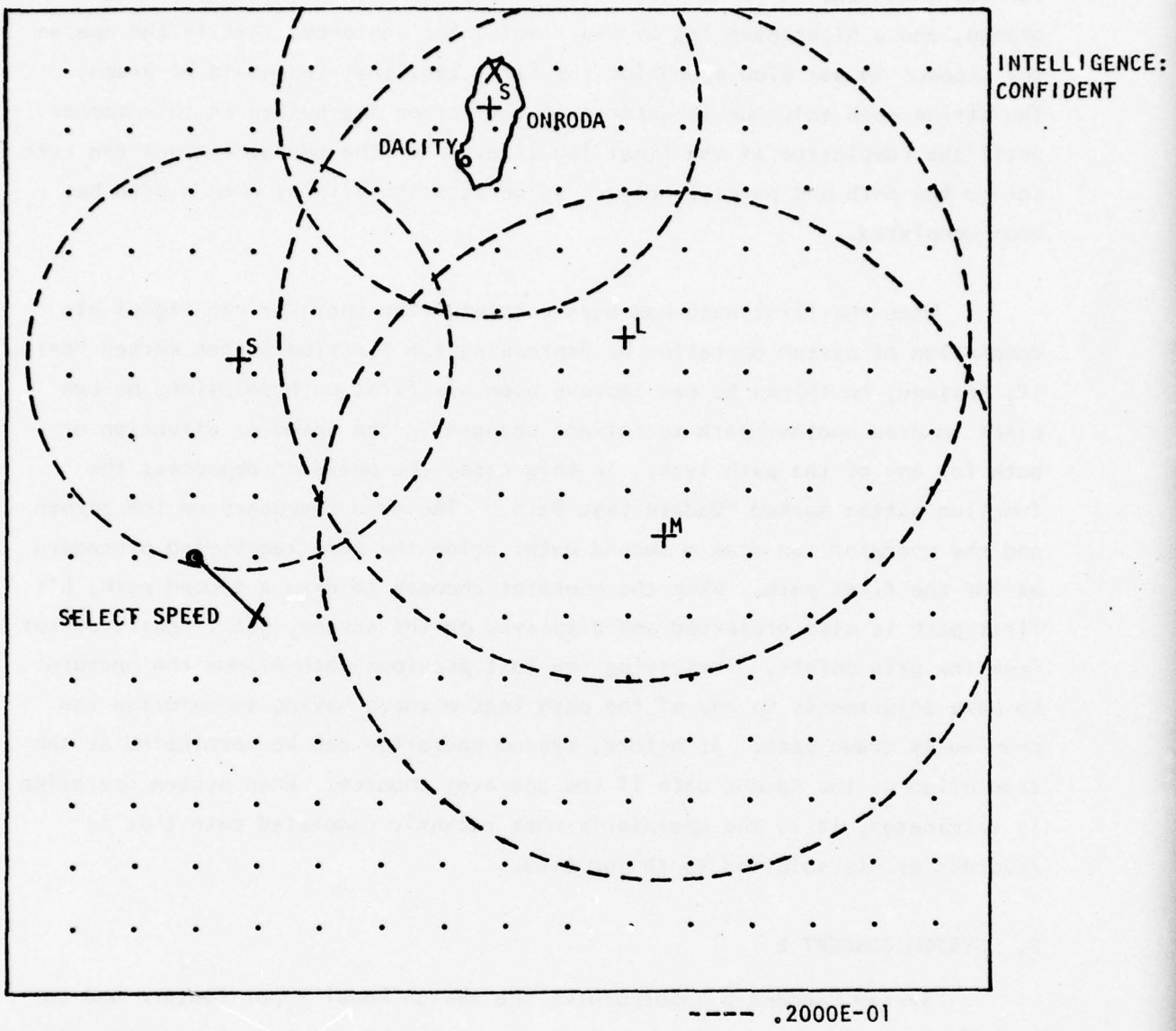


Figure 15. Display with One Path Leg Completed, System Concept A.

screen with the first leg of a strike path solution completed. Each leg of the strike path is color coded according to the speed selected by the operator for that leg. A "slow" leg is coded as green, a "medium" speed leg as orange, and a high-speed leg as red. Note, for instance, that if the operator chooses to use slow speed for the first leg, that leg would be green. The strike path solution is entered on the screen leg-by-leg in this manner until the completion of the final leg (i.e., when the operator moves the cursor to the path end point). Figure 16 depicts the display when a path has been completed.

When the first path has been completed the operator can signal his completion of system operation by depressing the function button marked "Exit." If, instead, he thinks he can improve upon his first path solution, he can elect to draw another path to reflect changes in the speed or direction or both for any of the path legs. In this case, the operator depresses the function button marked "Update Last Path." The grid reappears on the screen and the operator can draw a second path, using the same leg-by-leg procedure as for the first path. When the operator chooses to draw a second path, his first path is also preserved and displayed on the screen, but slightly offset from the grid points. Preserving the last previous path allows the operator to make adjustments to any of the path legs without having to memorize the previously drawn path. As before, system operation can be terminated at the completion of the second path if the operator chooses. When system operation is terminated, it is the operator's most recently completed path that is recorded as his solution to the problem.

B. SYSTEM CONCEPT B

System Concept B incorporates the Sketch Model input feature and certain feedback features. Under this concept, the operator analyzes the problem as it appears on the display and formulates a model of the composite detection rate field. He then makes his model explicit by entering it, via the Sketch Model procedure, on the display. He next solves for an optimal air strike path with respect to his inputted Sketch Model. Feedback is provided in two forms. One, the operator is shown the utility of each path he defines, enabling him to determine when he has optimized a strike path with respect to his Sketch Model. He is also provided a set of feedback histograms

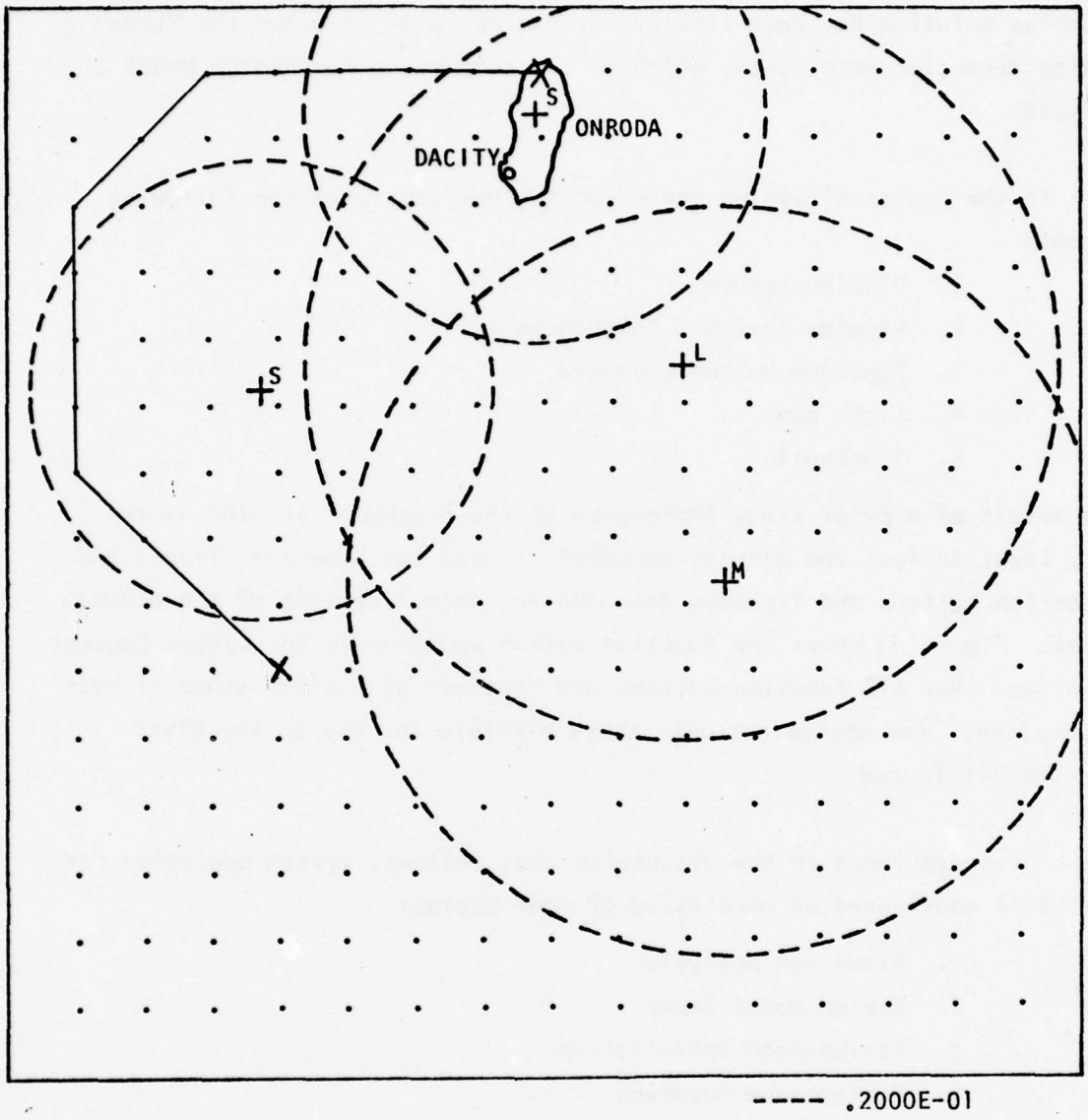


Figure 16. Display with One Path Completed, System Concept A.

that show the effects of each path leg chosen on the utility value components, which are fuel consumption and probability of being detected (according to his Sketch Model). Two, prior to the completion of system operation (but after the problem solution has been finalized), the operator is shown the "true" composite detection rate field, which he can compare to his Sketch Model of that field.

In the course of system operation the operator uses the following equipment:

1. Display screen
2. Display keyboard (alphanumeric)
3. Function button keyboard
4. Light pen
5. Trackball

As the result of a prior study (Reference 2) the trackball is used as the primary input device; the display keyboard is used for numerical input; and the function buttons and lightpen are used to control certain of the program features. Figure 17 shows the function button assignments for System Concept B operation. Not all function buttons are "active" at a given stage of system operation. The active buttons--those elibible for use at any given stage--are lit in red.

For convenience in the discussion that follows, system operation for Concept B is considered as consisting of four phases:

1. Situation Analysis
2. Sketch Model Input
3. Strike Path Optimization
4. Performance Feedback

1. Situation Analysis Phase

This phase of system operation is similar to that under Concept A, except that the operator has available a more detailed set of sensor coverage templates to support his detection field modeling effort. Elements of the display format and operator tasks for this phase are described below.

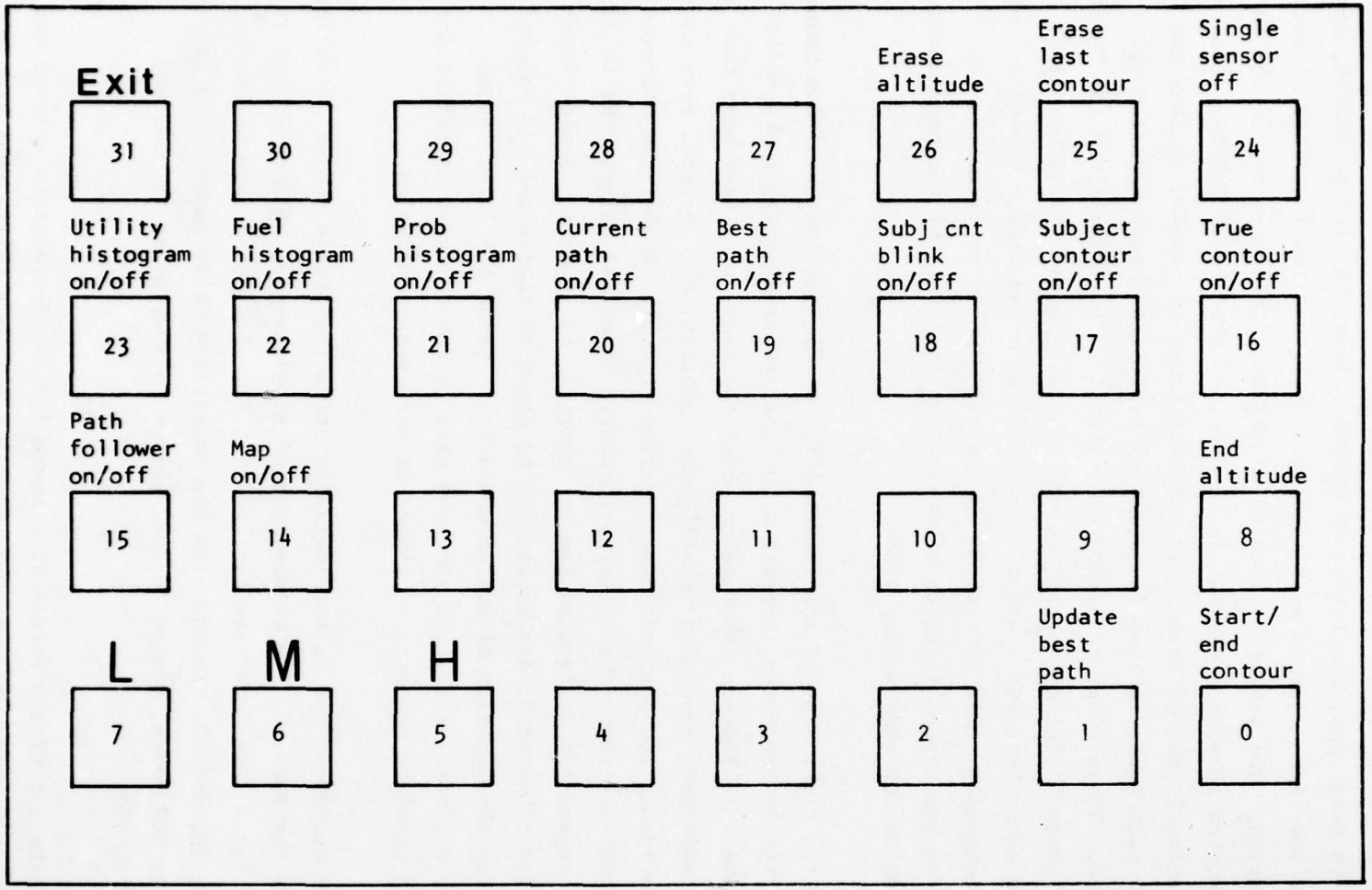


Figure 17. Function Button Assignments for System Concept B Operation.

a. Display Format. At the beginning of system operation the display screen appears as in Figure 18. The area within the boundary lines depicts the airstrike geography, consisting of ONRODA Island, DA City on ONRODA, an airstrip on the island (coded as a slanted cross), and the locations of each of four sensors. One sensor is located on ONRODA. The locations of the three platform-mounted sensors are shown in the "ocean" area below ONRODA. Each sensor location is marked by a cross and each cross is labeled to show the estimate of each sensor's type (in terms of detection range) reported by intelligence. These types are coded by letters: S for "Small," M for "Medium," and L for "Large." The operator does not associate numerical values with these three sensor ranges; instead he can call up "templates," scaled to the airstrike geography, that consist of concentric circles representing the distance from the sensor location of percentage intervals of the peak detection rate value for each sensor size.

At the right edge of the screen (Figure 18), outside the frame bounding the airstrike geography, appears a qualitative assessment of the reliability of the intelligence information about the sensors. There were four levels of confidence regarding intelligence information: perfect, very confident, confident, and not confident (Section II.A.4). Again, the operator does not associate these characterizations with numerical values. He is also free to disregard the designated sensor ranges if he chooses. Sensor locations, on the other hand, are assumed to be known perfectly and are modeled as static for the duration of system operation. Below the intelligence reliability estimate is a light pen menu used by the operator to select the size of the template he wants to appear at each sensor location.

The bottom of the screen initially shows the single sensor detection rate values for each of five percentages of single sensor peak detection rate value (0.2 for all sensors; see Section II.C.2). The percentage values are 90, 70, 50, 30, and 10. In addition the levels are color coded: red corresponds to the 90% value, orange to 70%, yellow to 50%, green to 30%, and dashed red to 10%.

At the top of the screen is a space for the operator to enter a numerical value representing his scalar subjective estimate of the peak composite detection rate. The procedure used to estimate this value is described below.

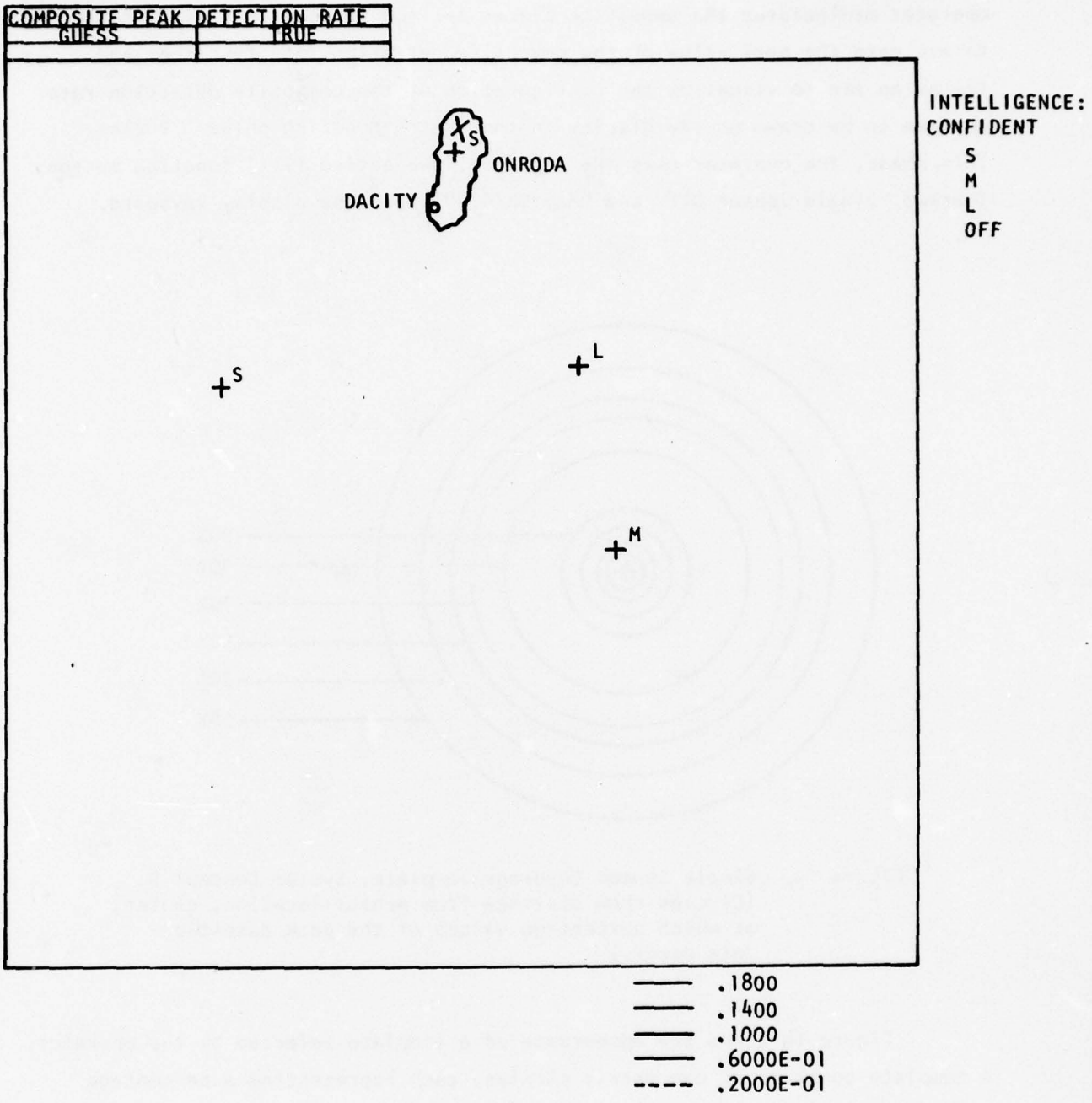


Figure 18. Display at Problem Initialization, System Concept B.

b. Procedure. During this first, situation analysis phase, the operator manipulates the sensor templates for two purposes: (1) as an aid to estimate the peak value of the composite detection rate function; and (2) as an aid to visualize the configuration of the composite detection rate surface to be drawn on the display in the Sketch Modeling phase. During this phase, the operator uses the lightpen, two active (lit) function buttons (marked "Single Sensor Off" and "Map On/Off"), and the display keyboard.

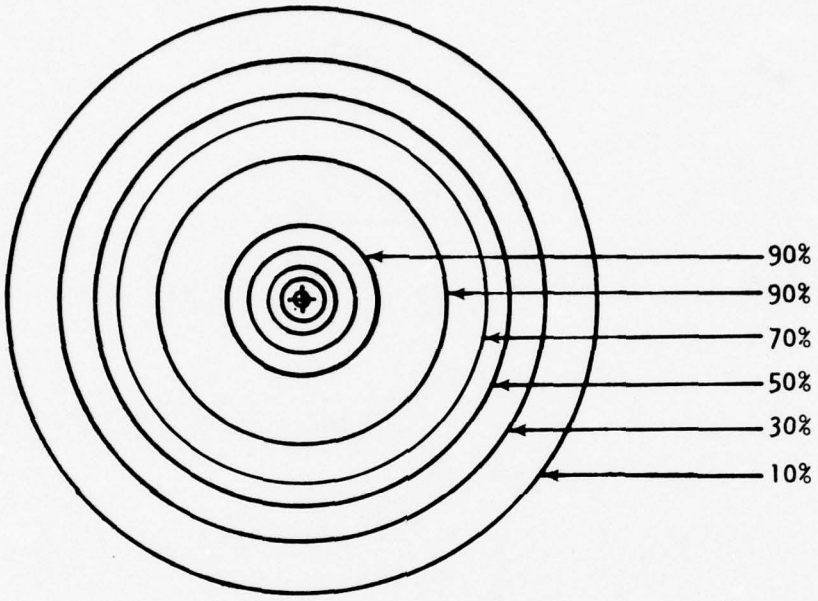


Figure 19. Single Sensor Coverage Template, System Concept B. (Circles show distance from sensor location, center, at which percentage values of the peak detection rate occur.)

Figure 19 shows the appearance of a template selected by the operator. A template consists of concentric circles, each representing a percentage value of the single sensor peak detection rate value. Each circle is color-coded according to the key for single sensor values displayed at the bottom of the screen. As noted above, the colors (red, orange, yellow, green, and broken red) correspond to 90, 70, 50, 30, and 10 percent respectively of the single sensor peak value of 0.2. The numerical values for each color-coded

contour are displayed at the bottom of the screen. Due to the "volcano" model of the single sensor detection rate function (Section II.C.1) each template has ten concentric circles: the inner five represent increasing percentages (the center is zero) and the outer five, decreasing percentages. There were three sizes of templates, small (S), medium (M), and large (L), representing different sensor types.

The operator uses the light pen to call up a template. Selecting any of the four sensor locations (coded as a cross), he points the light pen at the cross and depresses a button mounted on the light pen, causing that sensor cross to blink on and off. Next, the operator selects the size of the sensor coverage template he wants to overlay on the sensor location. As noted above, each sensor location has associated with it an intelligence-derived estimate of that sensor's coverage; the operator is free, however, to disregard the intelligence information. For instance, if intelligence suggests that a sensor had "medium" coverage (coded as "M" at the sensor location) but the reliability of the intelligence information for the trial is given on the right side of the screen as "Confident," the operator might decide instead to overlay a "Large" sensor coverage template on the sensor location. Once he decides the size of template he wants for the specified (blinking) sensor, he points the light pen at the appropriate letter in the menu at the right edge of the screen and presses the light pen button. This sequence of first activating (causing to blink) a sensor position, then selecting a template size from the menu, causes the template to appear on the screen centered about the sensor. Figure 20 depicts the display after one sensor coverage template has been called up.

The operator then repeats the sequence for each of the three remaining sensors. If he does proceed in this fashion, the display would appear as in Figure 21. The operator has several options, however, to allow him to manipulate the coverage templates. He can remove any of them from the screen, or he can change the size of any of them. To remove a template from the screen, the operator activates a sensor cross with the light pen, causing the cross to blink, then with the light pen selects the "Off" option from the menu at the right of the screen. To change a template size, he activates the sensor of interest and then selects another size from the menu. These

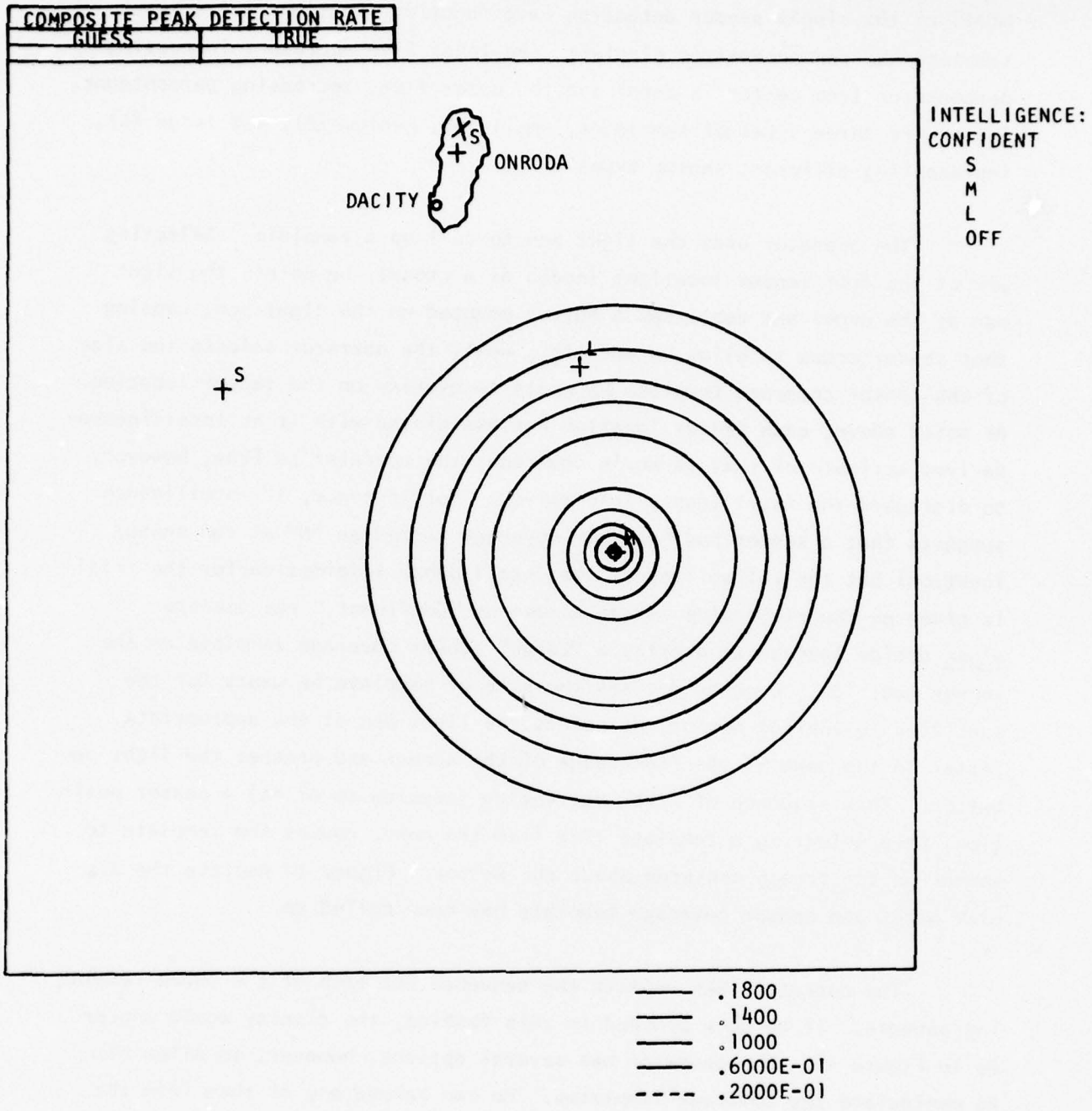


Figure 20. Display with One Sensor Coverage Template Selected, System Concept B.

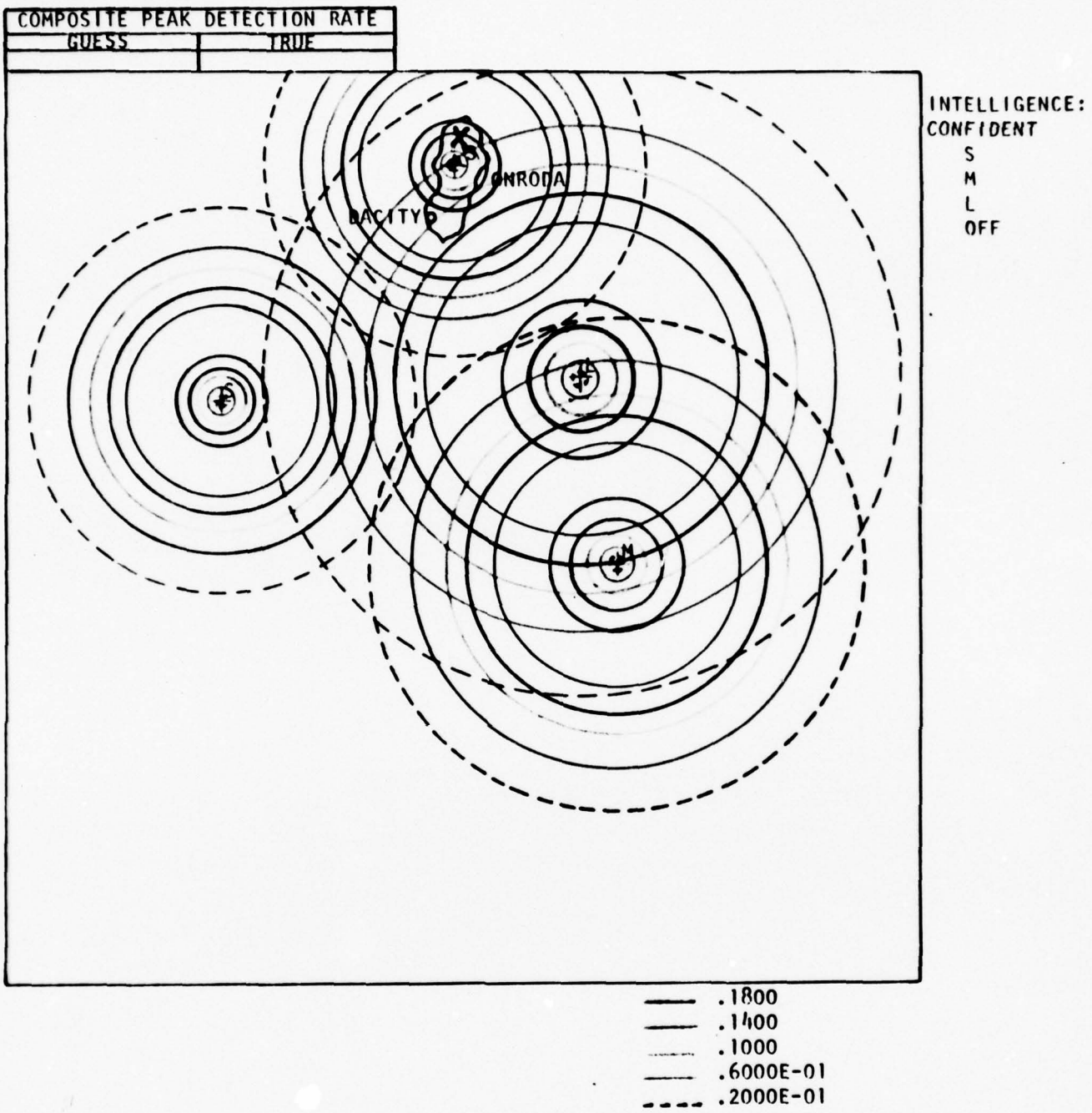


Figure 21. Display with all Sensor Coverage Templates Selected, System Concept B.

operations can be performed on only one sensor at a time. In addition, the operator can elect to remove ONRODA Island from the screen by pressing the lit function button marked "Map On/Off." The operator might choose this option to better study template overlaps in the area of the island. If he wants the map to reappear, he presses the same function button again.

There was no time limit placed on manipulating the sensor coverage templates, nor were there any rules for the operator to rely on to arrive at a "correct" combination of templates. At some point, however, he would be satisfied with his arrangement of templates. The display might then appear as in Figure 21.* At this point the operator studies the patterns of template overlap to discern the configuration of the composite detection rate function (to be Sketch Modeled on the display by the operator in the next phase of the trial) and to gauge the numerical value of the peak composite detection rate. Knowing that the composite surface results from simple addition of single sensor models, the operator can use the nature of template overlap in various regions to estimate the peak detection rate value.

When the operator decides that he has visualized the composite detection rate function and arrived at a reasonable estimate of the peak value of the composite function, he depresses the lit function button marked "Single Sensor Off." This action has three results: (1) the sensor coverage templates are erased from the screen; (2) the operator no longer has the templates available to him as an aid; and (3) the word "Guess" in the box at the top of the screen blinks, cuing the operator to enter the scalar value of his subjective estimate of the peak composite detection rate. At this point, the operator has no options: he is unable to call up the templates again, and he cannot continue with the trial until he enters his estimate. To do so, he uses the display keyboard to enter the decimal value of his estimate. Figure 22 shows the display as it appears after the subject cancels the templates and enters his peak value estimate. To move into the next phase of the trial, the operator activates the "Carriage Return" on the display keyboard.

---

*We have reproduced this figure in color to more fully represent the display format. Note that our color reproduction medium was insensitive to the color orange, so that blue was substituted.

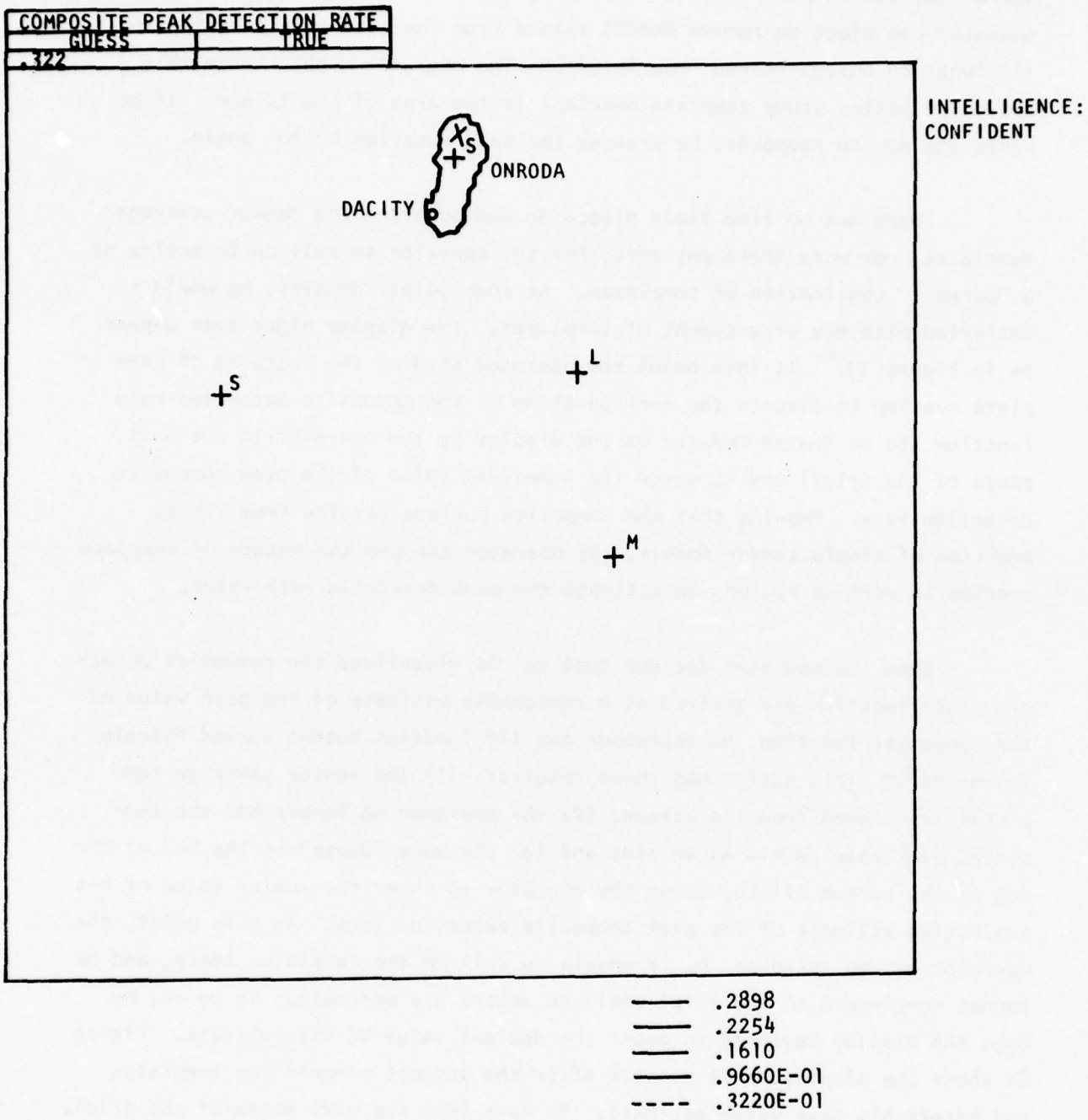


Figure 22. Display after Operator Estimate of Peak Detection Rate, System Concept B. (Note the absence of the sensor coverage templates and the display of the operator's scalar estimate of the peak composite detection rate.)

2. Sketch Model Input Phase

In this phase of system operation, the operator's task is to "draw," on the screen, a representation of his model of the composite multi-sensor detection rate function. He does this by drawing five iso-detection rate "contours," one for each "altitude" or percentage level of the multi-sensor composite function.* The detailed procedure follows.

a. Display Format. When the operator enters his numerical estimate of the peak composite detection rate and activates the "Carriage Return" of the display keyboard, the display screen format is immediately changed to that shown in Figure 23. The situation geography displayed within the frame remains, showing ONRODA Island and the sensor locations. The operator's estimate of the composite peak value remains visible at the top of the screen. The menu at the right no longer appears, but the intelligence reliability assessment remains. At the bottom right of the screen, the numerical values displayed now represent the percentages (90, 70, 50, 30, and 10%) of the peak composite value estimated by the operator. There is now a cursor that appears on the screen; during this phase, the operator manipulates the trackball to move the cursor producing the lines on the screen that represent the iso-"altitude" contours of his Sketch Model. At the lower left of the screen appears the legend, "Move Cursor." This legend blinks to cue the operator as to his next allowable action; at the start of this phase the red-coded line and its associated 90% value of the estimated peak composite detection rate also blink. At this point in the trial, the trackball is the only active display input device. The active (lighted) function buttons at the start of this phase of the trial are those marked "Map On/Off" and "Start/End Contour."

b. Procedure. By examining the overlaps of the sensor coverage templates in the preceding phase, the operator develops his estimate of the appearance of the composite detection rate function. His task during this phase is to transfer his mental image of it onto the screen. He does this by drawing color-coded contours representing the locations and shapes of the 90, 70, 50, 30, and 10% values ("altitudes") of his peak composite detection rate values.

---

*"Altitude" will be used throughout to mean the magnitude of the detection rate.

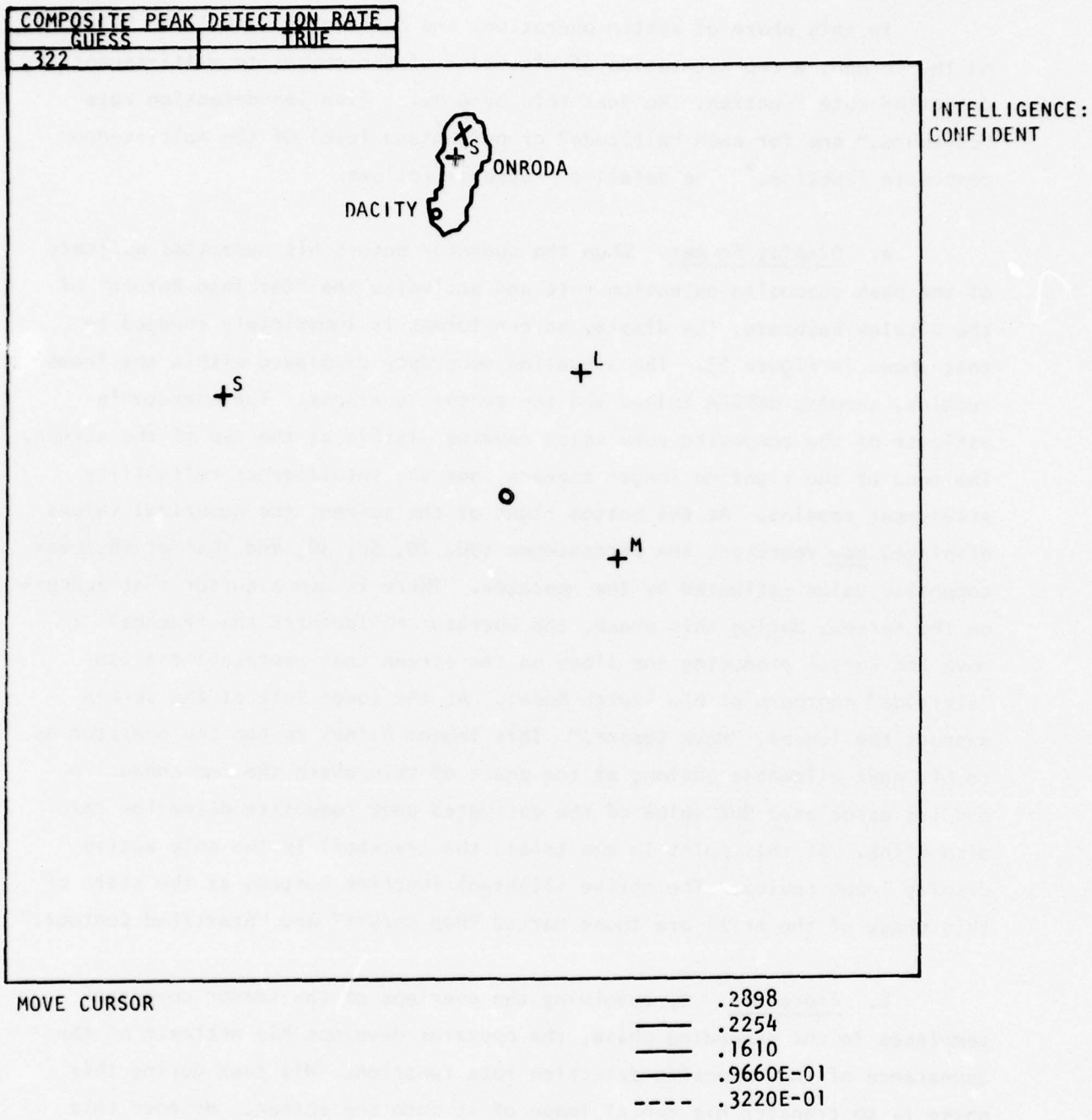


Figure 23. Display at Start of Sketch Model Input Phase, System Concept B. (Note the cursor positioned for drawing the first contour.)

There were a few rules established for this phase. First, all contours had to be drawn in the correct order. All contours for the 90% altitude were to be drawn first, followed by all contours for the 70% altitude, and so on through the 50, 30, and 10% altitudes. Second, at least one contour had to be drawn for each altitude. Third, each contour was required to be closed. Contours were not to be drawn outside the frame around the airstrike geography. In those cases where the templates showed that part of the composite function lay outside the frame boundary (as in Figure 21), only that portion inside the frame was to be drawn by the operator. Closure in this case was defined as both endpoints of the contour touching the frame. Finally, contours were to be drawn so that they did not intersect. The subject operators were instructed in these rules during their training. In addition, these rules were reinforced by having only those function buttons active at a time that led to allowable actions by the operator.

To perform this phase of the trial, the operator--cued by the blinking legend "Move Cursor"--uses the trackball to move the cursor to the desired starting point for the first contour at the 90% altitude. When he is satisfied with the cursor position, the operator presses the active function button marked "Start/End Contour." This legend blinks, as do the red line code and the 90% altitude value at the right. These cue the operator that he is now able to draw one contour, and that that contour is to represent the 90% altitude. Still using the trackball, the operator draws a contour in the shape and size he has decided represents a "peak" of the composite detection function. Figure 24 shows the drawing of the first contour in progress. The operator continues to close the contour. When the cursor is within a software specified minimum distance of the contour starting point, the contour is automatically closed.

Figure 25 depicts the display after the first contour has been drawn. The legend (bottom left) has returned to blinking "Move Cursor," and three additional function buttons have been activated. These are the ones marked "Erase Altitude," "Erase Last Contour," and "End Altitude." The four active function buttons, including "Start/End Contour," provide the operator with some options for his next action. If he thinks the composite surface is not unimodal, he can move the cursor and draw another contour at the 90% altitude

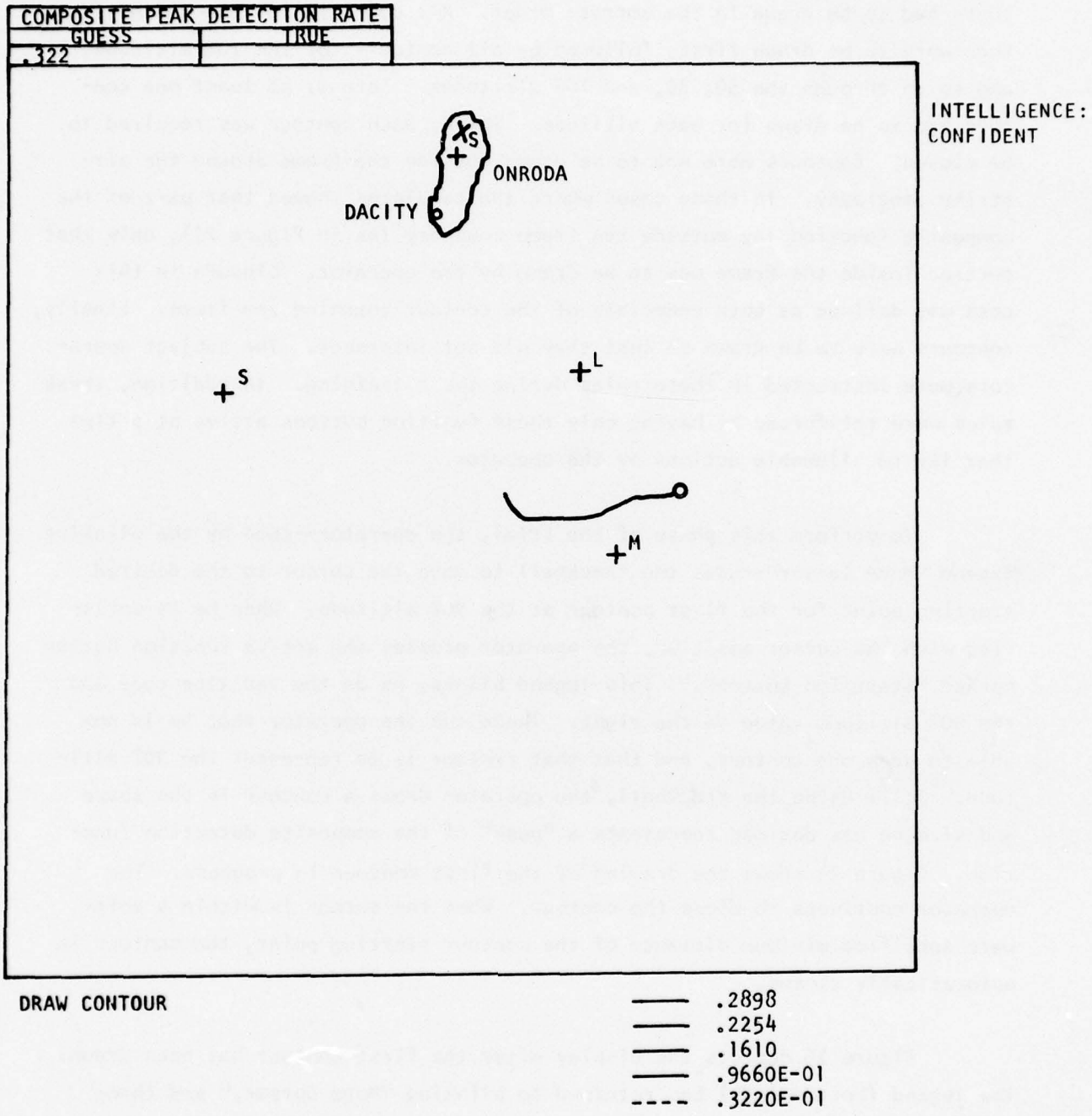


Figure 24. Display with First Sketch Model Contour in Progress, System Concept B.

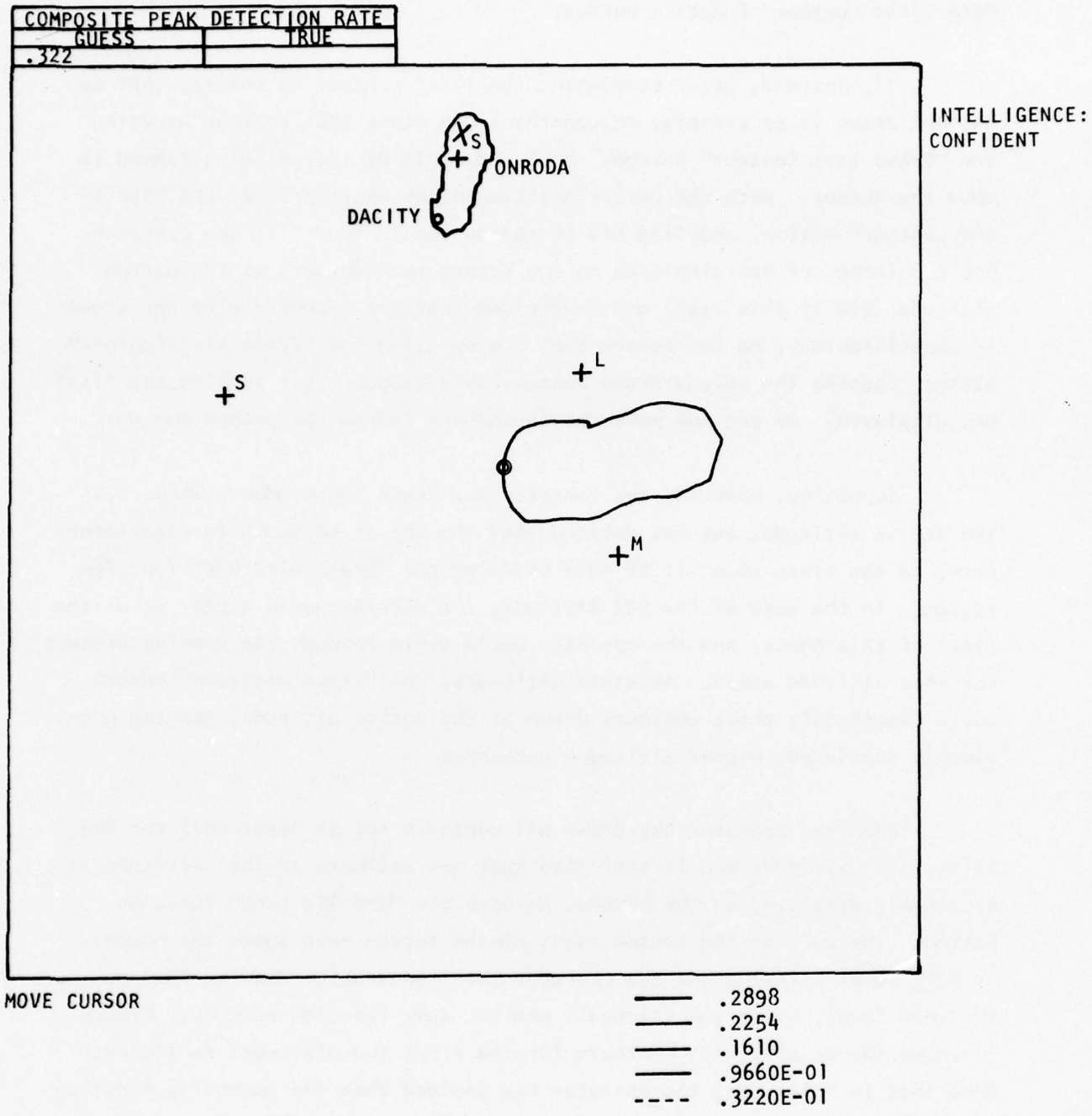


Figure 25. Display with First Sketch Model Contour Completed, System Concept B.

to represent the size, shape, and location of a second peak by engaging the "Start/End Contour" function button.

If, instead, after completing the first contour he decides that he has not drawn it accurately, the operator can erase that contour by using the "Erase Last Contour" button. He is now told by the blinking legend to move the cursor. With the cursor positioned, he again presses the "Start/End Contour" button, enabling him to redraw the contour. If the operator has completed and has displayed on the screen two contours at the current altitude (90% in this case) and he decides that the second one he has drawn is unsatisfactory, he can remove that one by using the "Erase Last Contour" button, causing the second-drawn contour to disappear, but leaving the first one displayed. He can now move the cursor and redraw the second contour.

Supposing, however, the operator has drawn two or more contours at the active altitude, but has decided that the set of contours is unsatisfactory, he can erase them all at once by using the "Erase Altitude" function button. In the case of the 90% altitude, the display would appear as at the start of this phase, and the operator would cycle through the drawing process for that altitude again. At other altitudes, the "Erase Altitude" button would cancel only those contours drawn at the active altitude, leaving previously completed, higher altitudes untouched.

When the operator has drawn all contours (or at least one) for the first (90%) altitude and is satisfied that his estimate of that altitude is accurately displayed on the screen, he uses the "End Altitude" function button. The code at the bottom right of the screen then shows the second, or 70%, level blinking and the operator then repeats the drawing sequence for that level, using the trackball and the same function buttons. Figure 26 shows the display with contours for the first two altitudes completed. Note that in this case, the operator has decided that the composite function is best represented by one "peak" at 90% and another at 70%. Figure 27 shows all five altitudes of the operator's estimate of the composite detection rate function represented. The complete set of contours thus comprises the operator's Sketch Model of the composite detection rate surface.

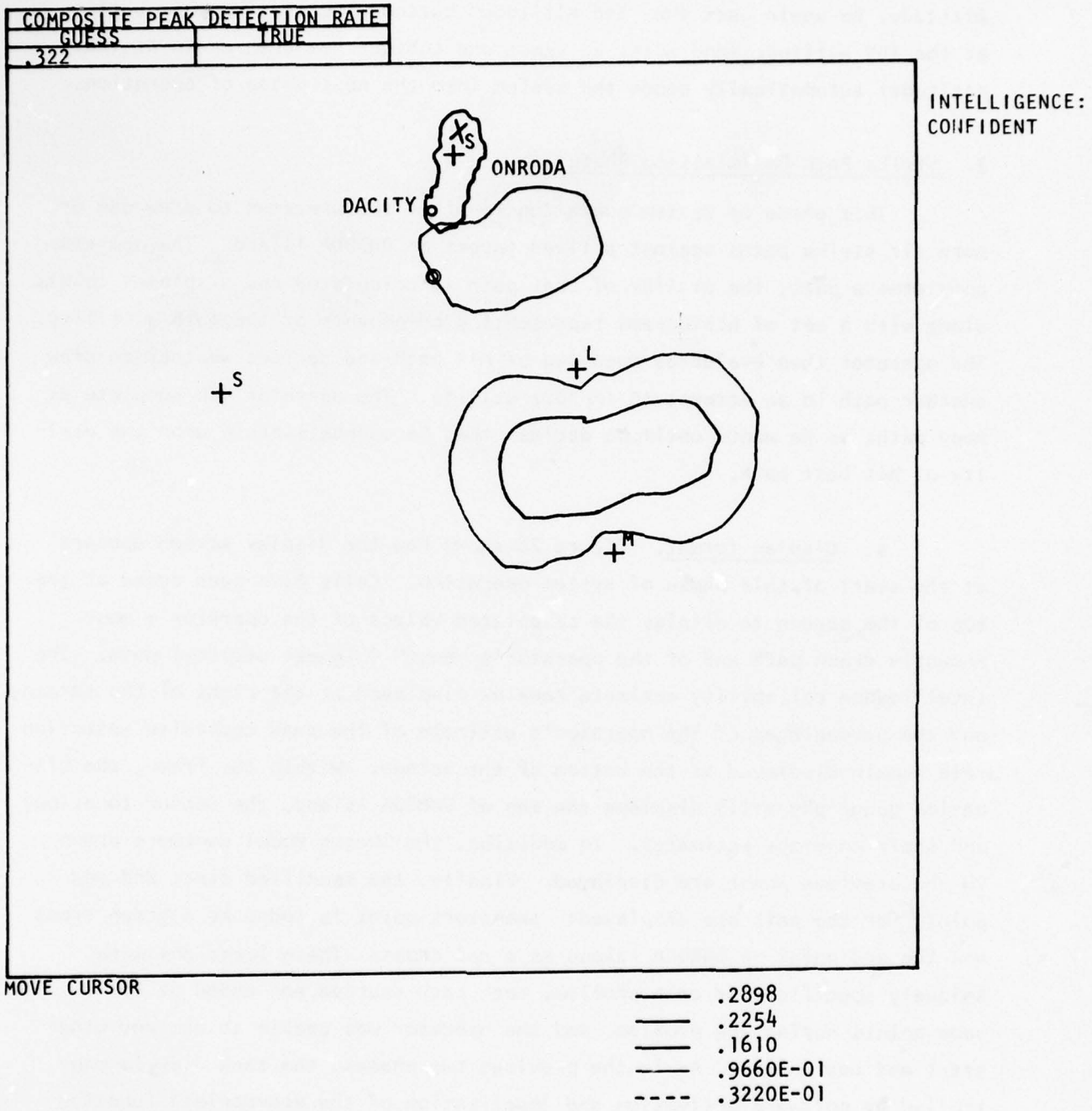


Figure 26. Display with Sketch Model Contours Completed for Two Altitudes, System Concept B.

Once the operator is satisfied with his contours for the fifth (10%) altitude, he again uses the "End Altitude" button. Activating this button at the 10% altitude (and after at least one contour has been drawn at that altitude) automatically sends the system into the next phase of operation.

3. Strike Path Optimization Phase

This phase of system operation requires the operator to draw one or more air strike paths against a fixed target on ONRODA Island. The operator completes a path; the utility of that path is calculated and displayed to him along with a set of histograms representing components of the path's utility. The operator then evaluates each leg of his path and decides whether to draw another path in an attempt to improve utility. The operator can complete as many paths as he wants until he decides that he cannot improve upon the utility of his best path.

a. Display format. Figure 28 shows how the display screen appears at the start of this phase of system operation. Cells have been added at the top of the screen to display the calculated values of the operator's most recently drawn path and of the operator's "best" (highest utility) path. The intelligence reliability estimate remains displayed at the right of the screen, and the percentages of the operator's estimate of the peak composite detection rate remain displayed at the bottom of the screen. Within the frame, the situation geography still displays the map of ONRODA Island, the sensor locations and their coverage estimates. In addition, the Sketch Model contours drawn in the previous phase are displayed. Finally, the specified start and end points for the path are displayed: the start point is coded as a green cross and the end point on ONRODA Island as a red cross. These locations were uniquely specified for each problem, each path started and ended at these same points during the problem, and the operator was unable to use any other start and end points. As in the previous two phases, the task flow is controlled by software activation and deactivation of the appropriate function buttons and by visual cues to the operator regarding his allowable actions at any point during this phase.

b. Procedure. When the display for this phase appears, immediately after the completion of the Sketch Model contours in Phase Two, the legend

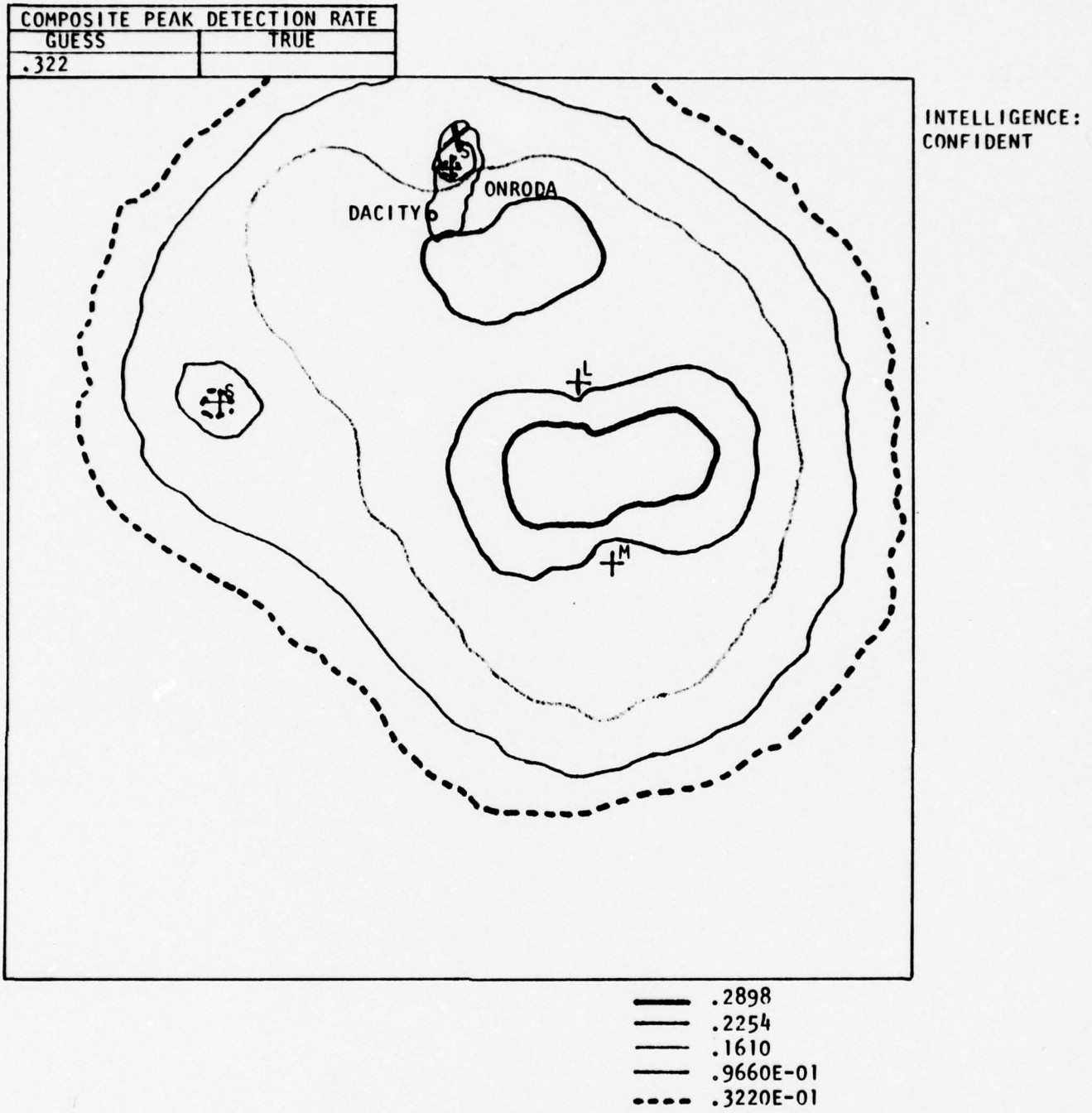


Figure 27. Display with all Sketch Model Contours Completed, System Concept B.

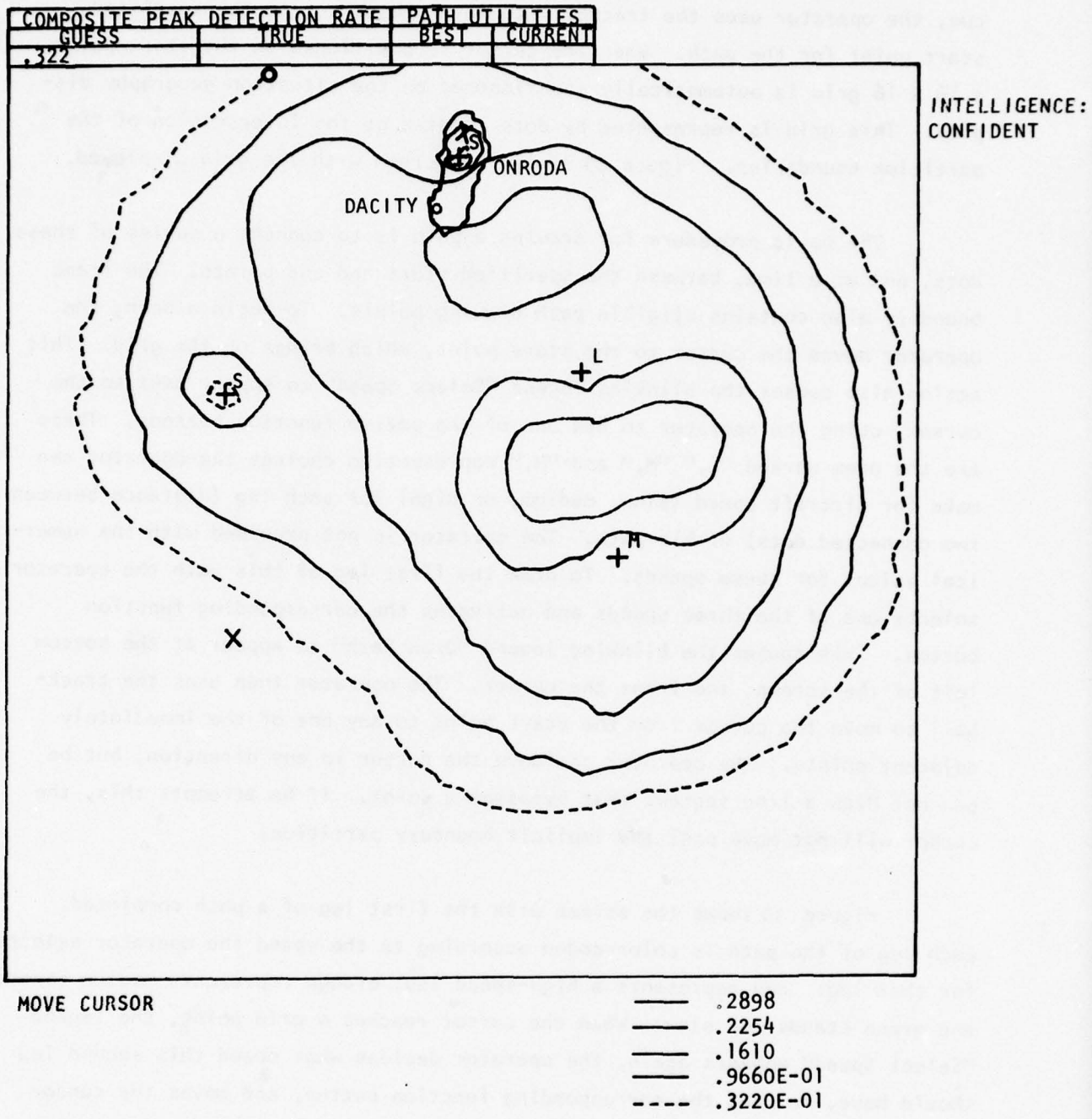


Figure 28. Display at Beginning of Strike Path Optimization Phase, System Concept B.

"Move Cursor" blinks at the bottom left of the screen. Responding to this cue, the operator uses the trackball to move the cursor to the specified start point for the path. When the cursor is positioned on the start point, a 16 x 16 grid is automatically superimposed on the situation geography display. This grid is represented by dots located at the intersection of the partition boundaries. Figure 29 shows the screen with the grid displayed.

The basic procedure for drawing a path is to connect a series of these dots, one at a time, between the specified start and end points. The frame boundary also contains eligible path drawing points. To begin a path, the operator moves the cursor to the start point, which brings up the grid. This action also causes the blinking legend "Select Speed" to appear next to the cursor, cuing the operator to use one of the active function buttons. These are the ones marked "L," "M," and "H," representing choices the operator can make for aircraft speed (slow, medium, or high) for each leg (distance between two connected dots) of his path. The operator is not provided with the numerical values for these speeds. To draw the first leg of this path the operator selects one of the three speeds and activates the corresponding function button. This causes the blinking legend "Draw Path" to appear at the bottom left of the screen, and frees the cursor. The operator then uses the trackball to move the cursor from the start point to any one of the immediately adjacent points. The operator can move the cursor in any direction, but he can not draw a line segment that bypasses a point. If he attempts this, the cursor will not move past the implicit boundary partition.

Figure 30 shows the screen with the first leg of a path completed. Each leg of the path is color-coded according to the speed the operator selects for that leg: red represents a high-speed leg; orange represents medium speed; and green stands for slow. When the cursor reaches a grid point, the legend "Select Speed" appears again, the operator decides what speed this second leg should have, he uses the corresponding function button, and moves the cursor to the next dot, completing the second leg. Each leg of the path is completed in this manner, by the operator first selecting speed and direction, then drawing that segment by connecting two dots. The final leg of the path is drawn to connect the red cross on ONRODA Island. An example of a completed path is shown in Figure 31.

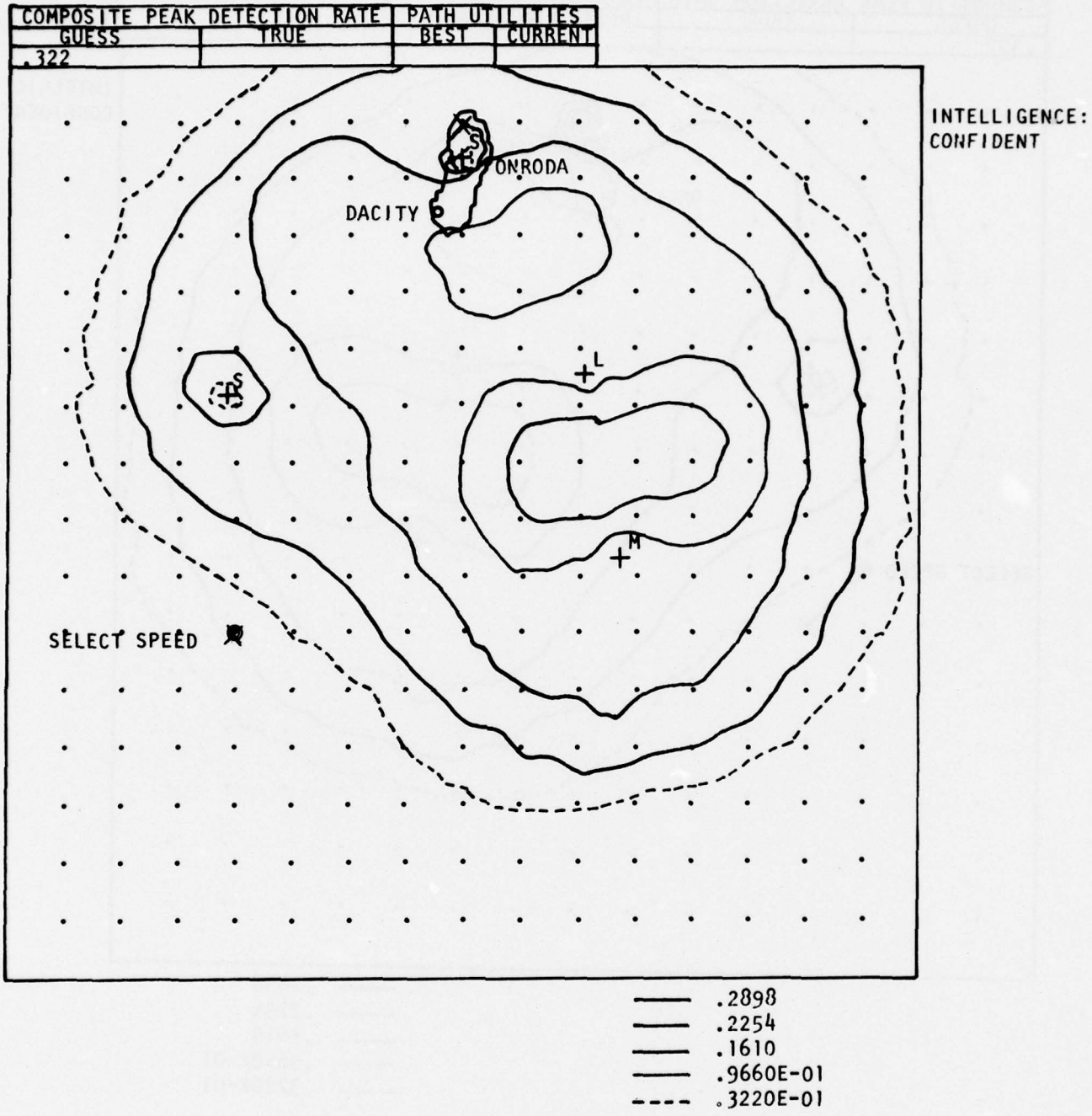


Figure 29. Display with Path Way Point Grid, System Concept B.

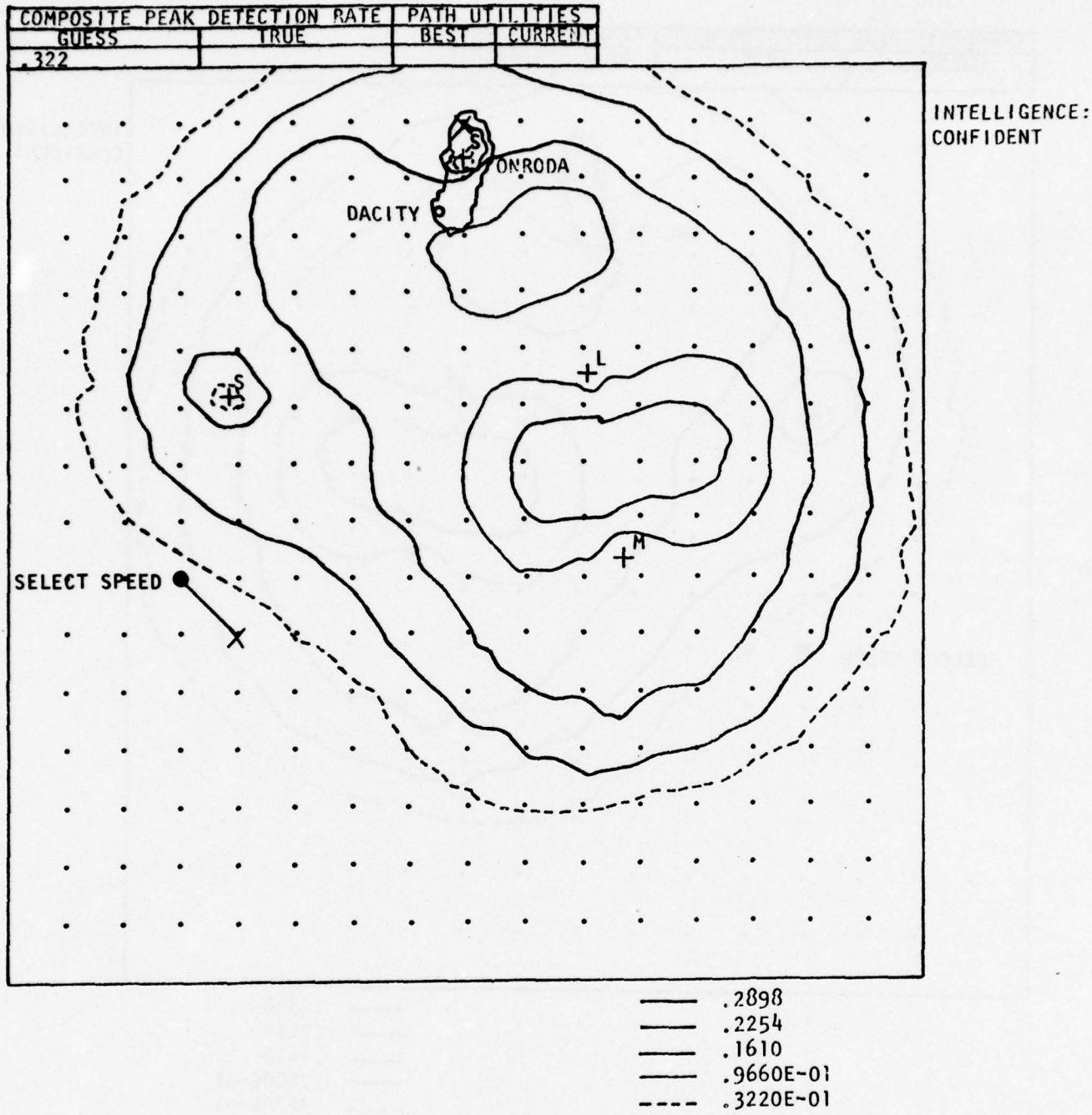


Figure 30. Display with First Path Leg Completed, System Concept B.

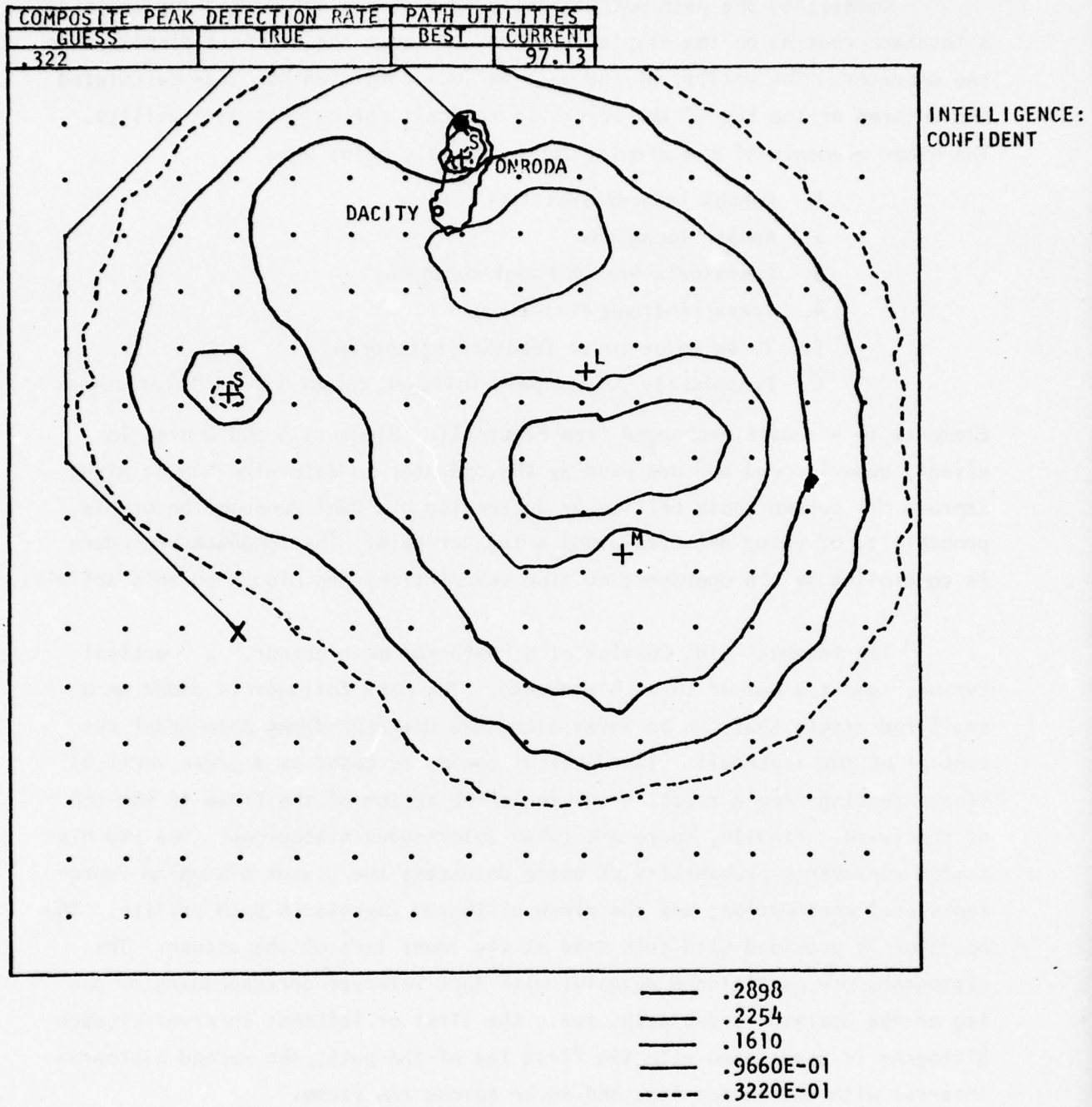


Figure 31. Display with First Path Completed, System Concept B. (Note path utility displayed in top right cell.)

Completing the path automatically removes the dot matrix and calls up a feedback routine on the display. Figure 32 shows the elements displayed to the operator. The utility of the path he just completed has been calculated and entered at the top of the screen in the cell for current path utility. The other elements of the display format at this point are:

1. ONRODA Island/airstrip
2. Sensor locations
3. Operator's Sketch Model contours
4. Operator-drawn strike path
5. Three color-coded feedback histograms
6. Dynamically paired path-follower cursor and vertical cursor

Elements 1 - 4 remain unchanged from Figure 31. Elements 5 and 6 are, in effect, superimposed and are used by the operator to determine how he might improve his current path utility by decreasing his fuel consumption or his probability of being detected along a leg, or both. The feedback procedure is controlled by the operator; no time restrictions are placed on this activity.

The feedback aids consist of a "path-follower cursor," a "vertical cursor," and the set of three histograms. The path follower is coded as a small red circle that can be moved along the operator-drawn path under the control of the trackball. The vertical cursor is coded as a green vertical line extending from a tracking circle at the bottom of the frame to the top of the frame. Finally, there are three color-coded histograms. The red histogram represents probability of being detected; the orange histogram represents fuel consumption; and the green histogram represents path utility. The operator is provided with this code at the lower left of the screen. The histograms are partitioned equally, with each interval corresponding to one leg of the operator-drawn path, i.e., the first or leftmost interval of each histogram is associated with the first leg of the path, the second histogram interval with the second leg, and so on across the frame.

To use this feedback information, the operator rolls the trackball to move the path follower along the path. Moving the path follower also moves the vertical cursor as an index, so that the operator can "read" the histogram levels corresponding to any or all legs of his path. For instance, Figure 32.

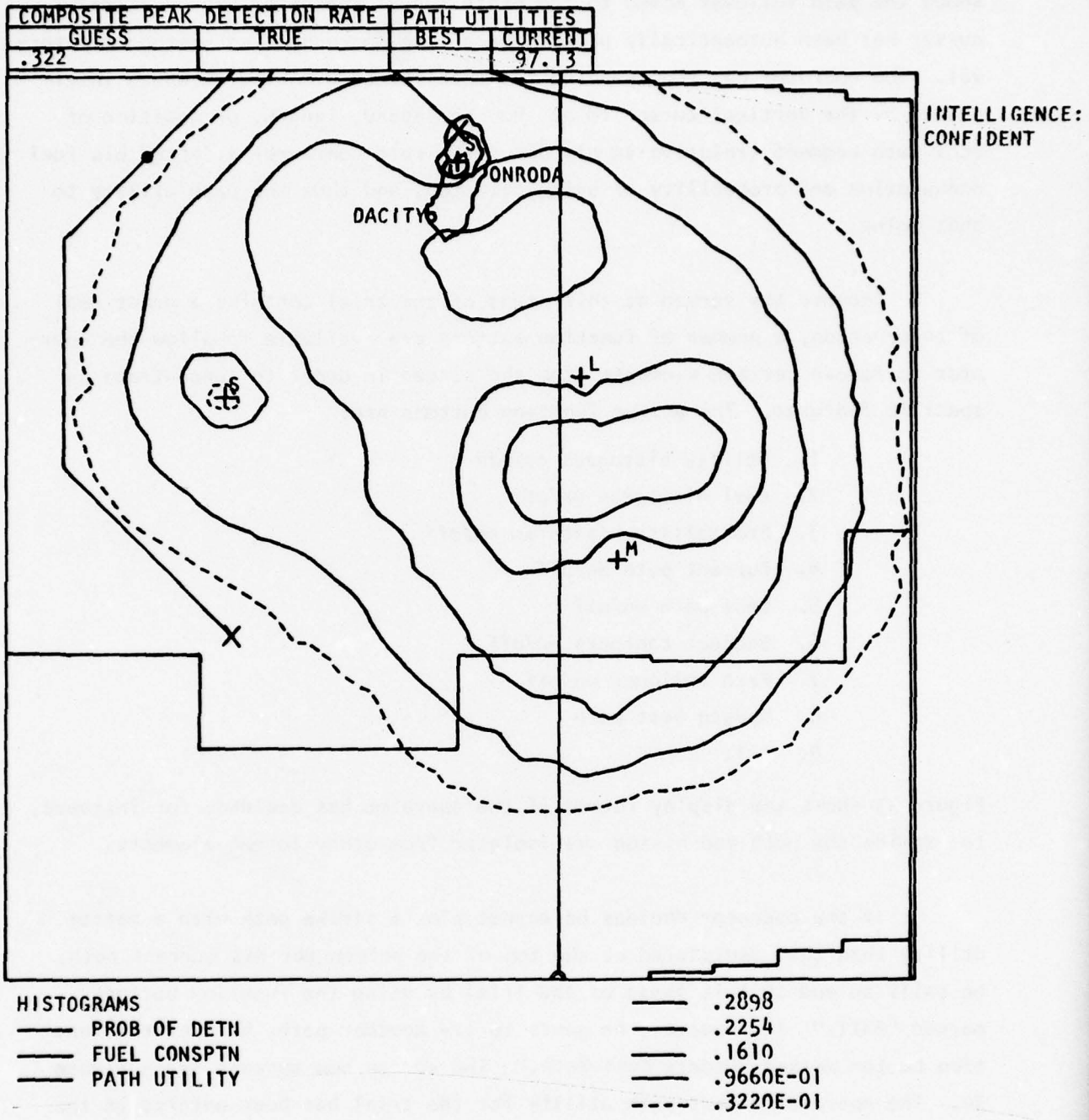


Figure 32. Display with Path Feedback Histograms, System Concept B.

shows the path follower moved to the ninth leg of the path. The vertical cursor has been automatically positioned at the corresponding histogram interval. The operator can then examine the level of each histogram where intersected by the vertical cursor to see how the speed, length, or position of that path segment (relative to his detection rate contours) affected his fuel consumption and probability of being detected, and thus the path utility to that point.

Because the screen at this stage of the trial contains a great deal of information, a number of function buttons are available to allow the operator to remove certain elements from the screen in order to concentrate on specific features. The active function buttons are:

1. Utility histogram on/off
2. Fuel histogram on/off
3. Probability histogram on/off
4. Current path on/off
5. Best path on/off
6. Subject contours on/off
7. Path follower on/off
8. Update best path
9. Exit

Figure 33 shows the display format if the operator has decided, for instance, to examine the path and histograms isolated from other format elements.

If the operator decides he cannot plot a strike path with a better utility than that registered at the top of the screen for his current path, he calls an end to this phase of the trial by using the function button marked "EXIT." If, however, he wants to try another path, he uses the function button marked "Update Best Path." The screen now appears as in Figure 34. The operator's best path utility for the trial has been entered in the appropriate cell at the top of the screen. (If the operator has drawn only one path so far in the trial, best path utility will be that one; if he has drawn several, best path utility will be the greatest one to date.) The dot grid reappears and the operator's best path is preserved so that he can refer to it if necessary; it is displayed slightly offset from the grid so

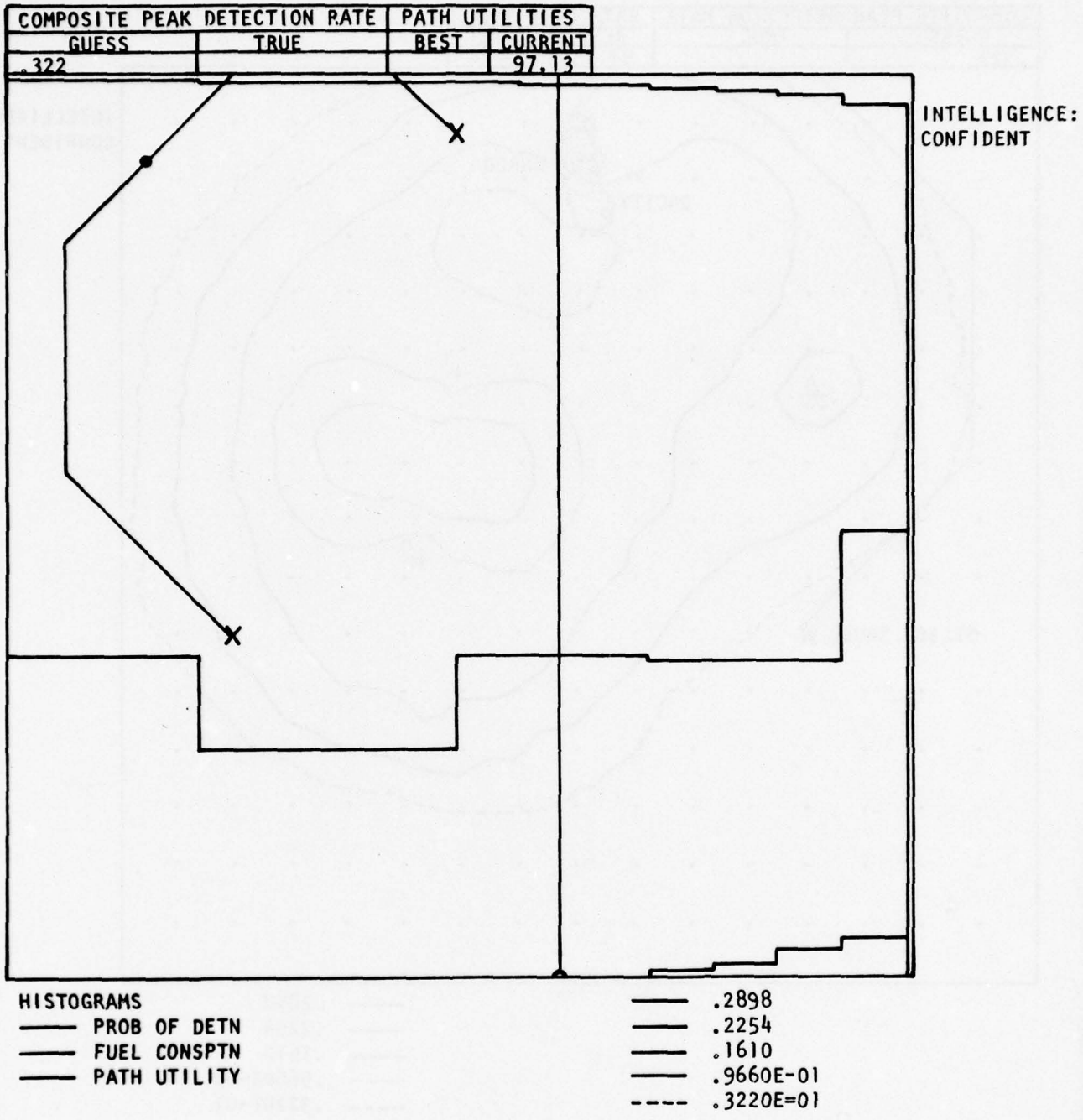


Figure 33. Display with Feedback Histograms Only, System Concept B.

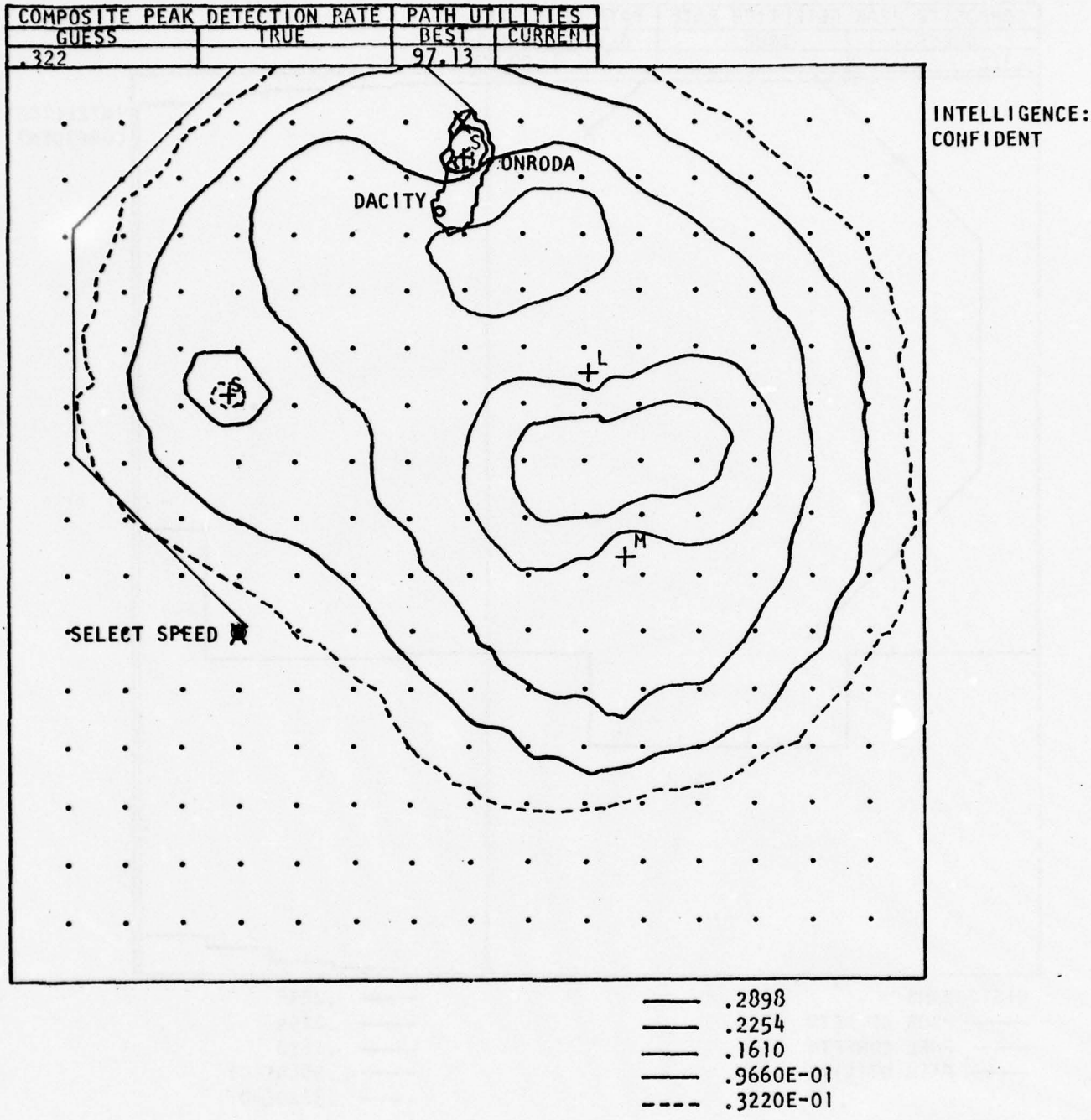


Figure 34. Display at Start of Second (optional) Path, System Concept B.

that it does not interfere with drawing another path. The operator draws another path (using the same start and end points) according to the procedure described above. At the completion of the path the display is returned to the feedback format described above. Feedback histograms appear for the just-drawn path. In addition, the utility for the new path has been calculated and entered in the cell for current path utility at the top of the screen.

The operator continues with this phase until he decides he can not improve upon his best path utility. At that point he signals his completion of this phase by using the "EXIT" function button.

4. Composite Detection Rate Function Feedback Phase

The final phase of the trial was provided to enable the operator to compare his composite detection rate function (represented by the Sketch Model contours he drew in Phase Two of the trial) with the "true" composite detection function for that trial. No operator inputs were elicited during this phase.

a. Display Format. When the operator exits from the path optimization phase, the Sketch Model contours he has drawn start to blink and the "true" contours are automatically drawn on the screen, one altitude at a time, starting with the 90% level, and with each altitude level color coded. The best (highest utility) path from the preceding phase is updated and remains on the display; its path utility is displayed at the top of the screen. In addition, the true peak composite detection rate for the trial is automatically entered next to the operator's estimate at the top of the screen. Figure 35 shows the display format for this phase, after all the true contours have appeared.

b. Operator Activities. Since this phase is used by the operator for feedback only, the operator is provided options to unclutter the display format, allowing him to examine the configuration of the true contours and to compare them with the Sketch Model contours he has drawn. These options are controlled by the following active function buttons:

1. Best path on/off
2. Subject contour blink on/off
3. Subject contour on/off
4. True contour on/off
5. Map on/off
6. Exit

Selective use of these function buttons (except for the "Exit" button) allows the operator to see various combinations of the display format elements. If he wishes to see only the true contours, for instance, he can turn off his path, his contours, and the map, causing the display to appear as in Figure 36. To examine his best path in conjunction with the true contours, he uses the appropriate function button to turn his path back on (Figure 37). To compare his contours with the true contours, he might turn his path off again and turn on his contours (Figure 38). When the operator has examined the combinations of display elements meaningful to him, he presses the "Exit" function button, ending system operation.

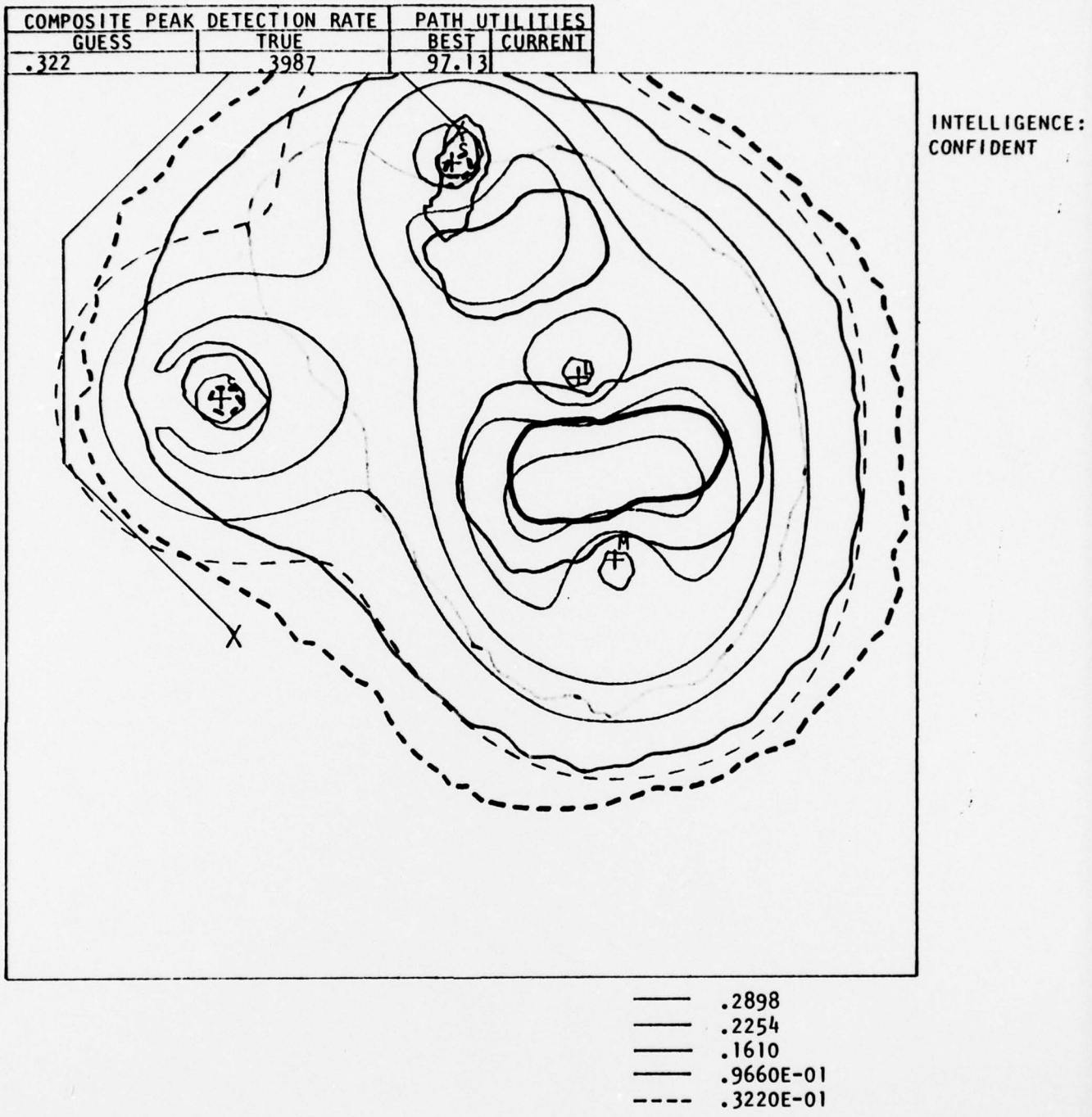


Figure 35. Display with True Feedback Contours (shown here in black) Superimposed on Subject's Sketch Model Contours (color coded), System Concept B.

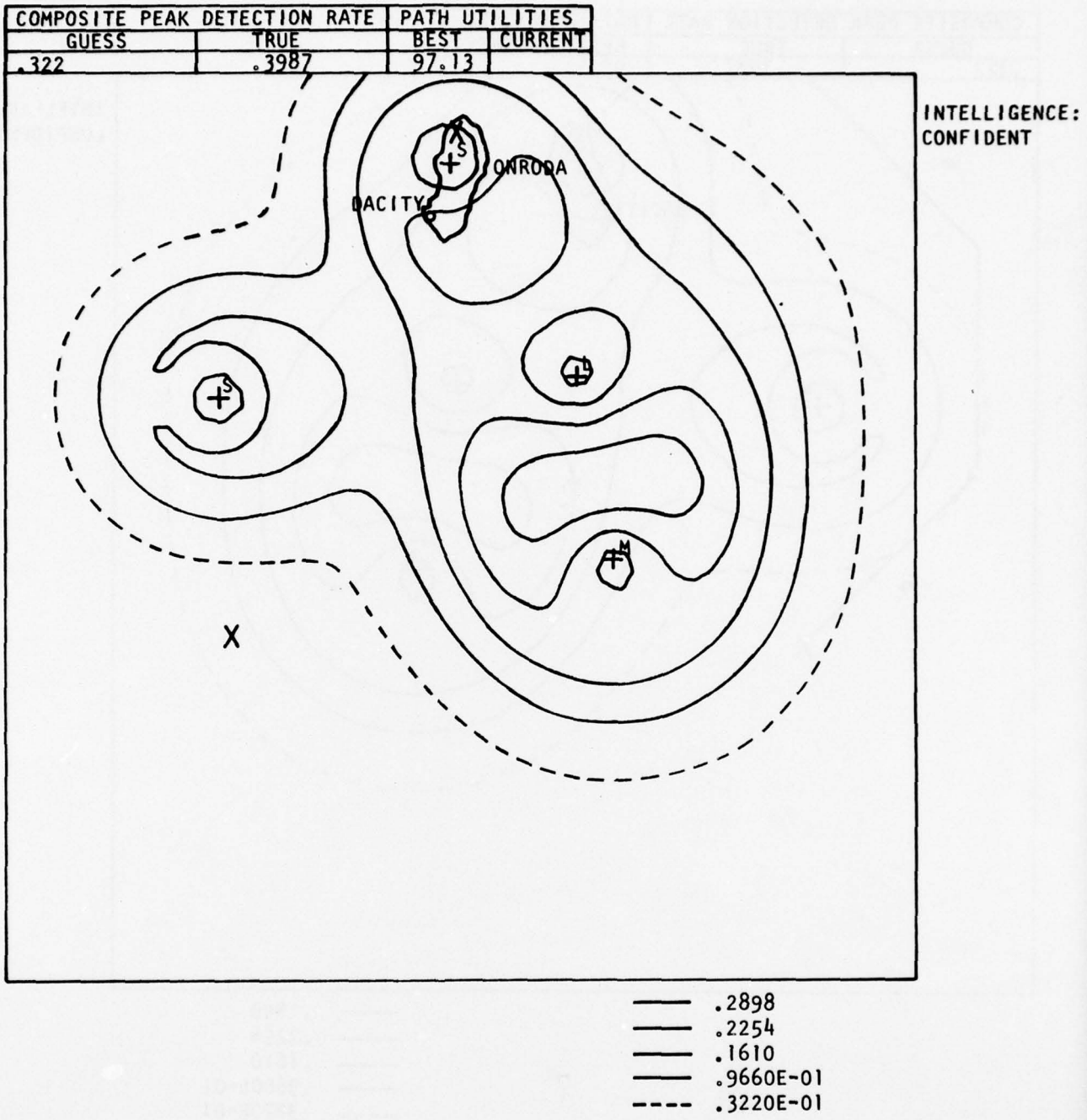


Figure 36. Display with True Contours Only, System Concept B.

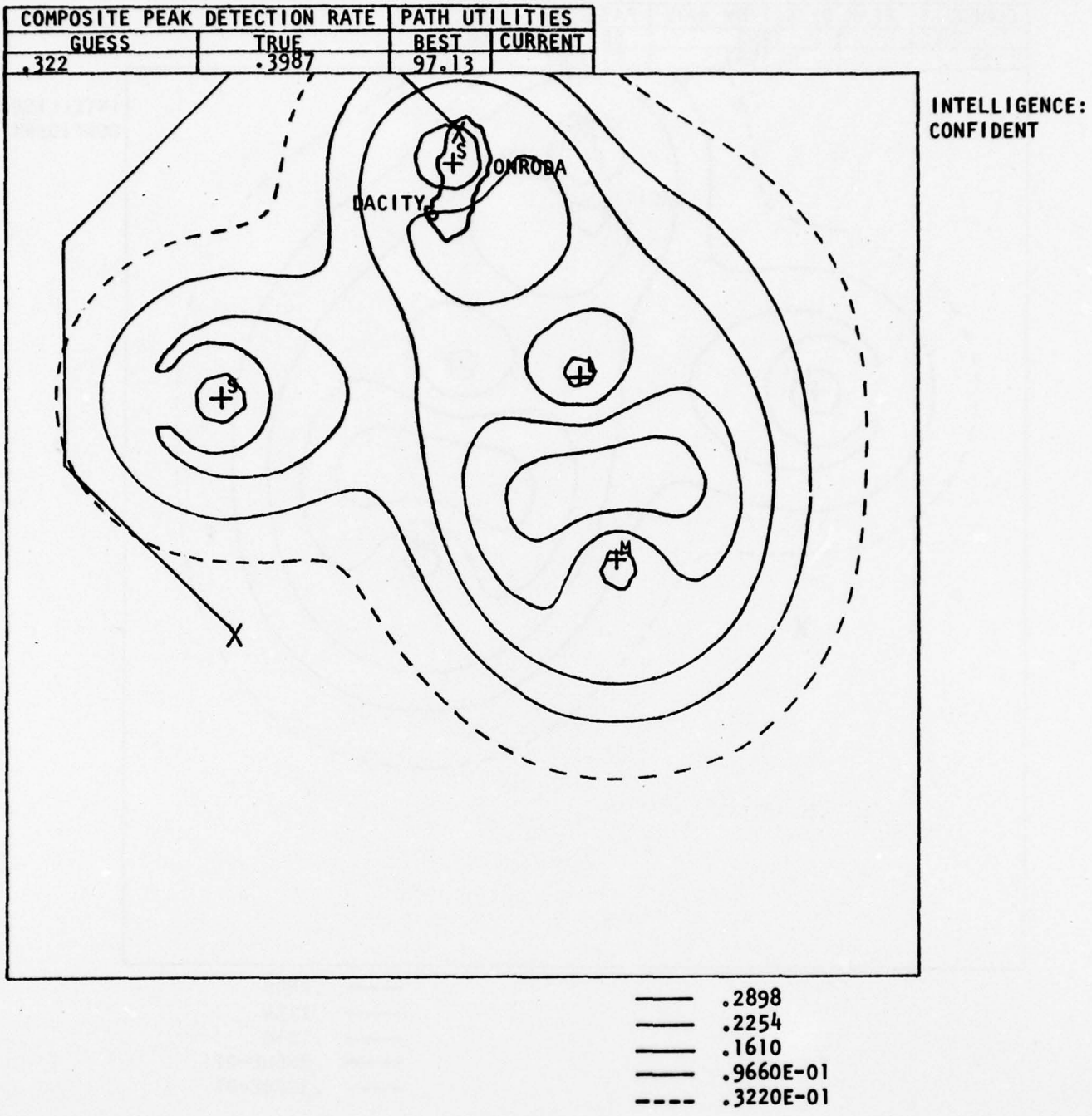


Figure 37. Display with True Contours and Subject's Best Path, System Concept B.

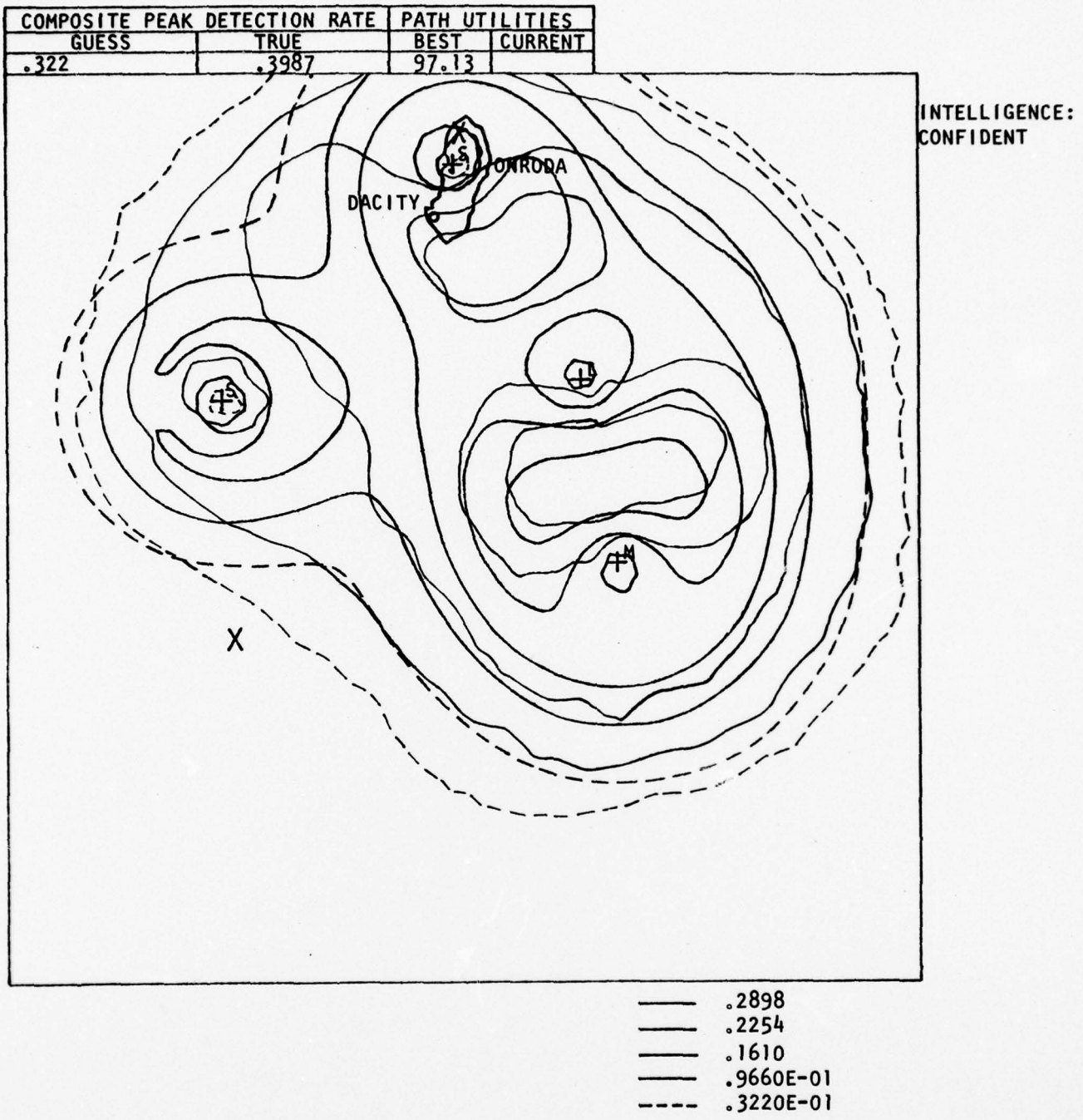


Figure 38. Display with True and Sketch Model Contours Only, System Concept B.

IV. DESCRIPTION OF THE EXPERIMENTS

A. EXPERIMENTAL DESIGN

1. Hypotheses

There were three experiments. Two experiments involved the collection of strike path utility data. In one experiment the strike path optimization function was carried out under System Concept A by two groups of operators, with one group experienced and the other inexperienced in the Sketch Model technique. Although the Sketch Model approach was not implemented under System Concept A, it was necessary to determine whether the performance of the group experienced in Sketch Modeling differed from that of the naive group. Accordingly, the experimental null hypothesis is:

> Path utilities generated by the experienced and naive operator groups are not significantly different as a function of groups, problem difficulties, replications, or their interactions.

Implicit in this hypothesis is the question of whether the group experienced in Sketch Modeling performed better than the naive group. In other words, it was necessary to determine whether prior experience with Sketch Models biased performance when the same task was done, but without the use of Sketch Models. If not, then the utility data generated by the experienced operators on Concept A could be compared with the data they generated under Concept B.

Given that the utility data generated by experienced operators using Concepts A and B can be validly compared, then the other experiment was designed to compare path utilities across Concepts A, B, C, and D. Concept D cannot be compared directly with the other concepts by analysis of variance because there are no operators for Concept D and the utility data are not independent random variables. Therefore, two experimental null hypotheses are needed:

1. Path utilities for Concepts A, B, and C are not significantly different as a function of concepts, operators, problem difficulties, replications, or their interactions.
2. Path utilities for Concepts A, B, C, and D are not significantly different as a function of system concepts.

Implicit within these hypotheses are the following questions:

1. Is there a difference between performance under Concepts A and B? That is, does the use of Sketch Model contours (Concept B) help the operator to devise paths with utilities significantly better than those generated without the use of Sketch Model contours (Concept A)?
2. Is there a difference between performance under Concepts B and C? That is, does the use of the dynamic programming optimization method to select strike paths based on Sketch Model contours help to achieve paths with significantly better utilities than those generated by operators using the same contours?
3. Is there a difference between Concepts A and C? That is, does the use of the dynamic programming optimization method to select strike paths based on Sketch Model contours yield paths with significantly better utilities than those generated by operators who do not use Sketch Model contours?
4. Is there a difference between Concepts C and D? That is, does the use of the dynamic programming optimization method to select strike paths based on the true detection rate contours yield significantly better utilities than those generated by the same optimization operating on the Sketch Model contours?

A third experiment, involving collection of Sketch Model accuracy data, was performed. The experimental null hypothesis is:

> Errors in Sketch Model accuracy are not significantly different as a function of operators, problem difficulties, replications, or their interactions.

Acceptance or rejection of this hypothesis sheds light on the source of variations in the errors. In order to evaluate the accuracy of the contours themselves, a separate analysis was performed on the data so that subjective inferences could be made about the adequacy of Sketch Model contours for applications other than development of strike paths as investigated in the experiment.

2. Independent Variables

The independent variables for the experiments were:

a. Subjects/Subject Groups
b. System Concepts
c. Replications
d. Problem Difficulty

a. Subjects/Subject Groups. Subject operators for this study were treated as experimental variables in two ways. For two of the experiments each subject was considered as a variable. As can be seen from the ANOVA designs (Section IV.A.5, below), four individuals were used in conjunction with the experiment that compared System Concepts A, B, C, and D, and with the experiment on Sketch Model accuracy. In the third experiment, subject groups were the variables. There were two groups of four subjects each. One group had received prior training in the Sketch Model procedure, and the other group was untrained in the Sketch Model procedure. The two subject groups were considered variables for the portion of the analysis that focused on performance under System Concept A. Each of the eight subjects was a U.C.L.A. undergraduate. Each had a technical major field (engineering, computer science, and psychology majors were used), and hence some analytical background. None had military experience.

b. System Concepts. Each of the four system concepts described in Section II.B. was used in the experimental design. To review, the system concepts differed primarily in the way the two major functions, modeling and optimization, were allocated. Concepts A and B both allocated the two functions to the operator; under Concept A, the operator's mental model remained implicit, while under Concept B, the operator's mental model was made explicit and stored in the computer via the Sketch Model procedure. Concept C assigned the modeling function to the operator and the optimization to the computer. Concept D allocated both functions to the computer.

c. Replications. Ideally, the learning effect is tested by giving each subject two or more identical sets of problems and by giving each subject the identical sets of problems. In the experiment reported here, each subject solved the same set of 48 problems. However, there were not identical sets of problems within the 48 problems solved by a subject. The reason was the concern that a subject's response to the second occurrence of a problem might be influenced by his memory of the solution chosen at the first occurrence.

The procedure adopted was to formulate 12 problems within each of the four difficulty levels. Each of the 12 problems within each of the difficulty

levels that included uncertainty of intelligence information was selected at random. Each subject's trials were arranged so that he had six problems from each difficulty level in each half of the 48 trials. A replication, then, consisted of 24 problems with 6 in each difficulty level. Since the problems in a difficulty level were not identical for each replication, the replications factor is not the ideal measure of the learning effect. However, by selecting the problems within each difficulty level at random and using equal numbers of problems within each difficulty level in each replication, we have as good measure of the learning effect as could be obtained within the constraint of 48 trials per subject.

d. Problem Difficulty. As an experimental variable, problem difficulty pertains only to the intelligence reliability assessment. There were four levels of confidence in the reliability of intelligence information: perfect, very confident, confident, and not confident. The least difficult problems with respect to this variable, therefore, were those with perfect intelligence information; the most difficult were those for which intelligence was assessed as "not confident."

Other factors were varied from problem to problem in an effort to assure non-trivial problems, thus, in a sense contributing to problem difficulty. These factors, however, were not treated as variables in the experimental design. Rather, they were considered "scenario variables" and are discussed as such in Sections IV.A.4 and VI.B.

3. Dependent Variables

a. Sketch Model Accuracy. This section describes an algorithm that provides a quantitative measure of similarity between the detection rate surfaces represented by Sketch Models under System Concept B and the true surfaces generated by Concept D. The procedure devised involves sampling both surfaces and combining the measurements to form a single measure of Sketch Model accuracy. The procedure below was carried out for each operator's Sketch Model for each problem.

Both surfaces are three dimensional, with the z-direction being the detection rate and the x- and y-directions specifying the geographical position.

The true surface is continuous in the z-direction while the Sketch Model surface is discrete. Figure 39 shows the true composite function superimposed over a sampling grid; Figure 40 shows the Sketch Model, or discrete, detection surface over the same grid.

By sampling each surface at the grid points, the height of each can be determined. We can then determine the difference in the altitudes of the two surfaces at each grid point. Figure 41 shows the result of taking one cross-section through each of Figures 39 and 40 at the indicated x_i. At each y_j (represented by tic marks) along this i^{th} slice we can determine the difference in altitude between the two surfaces. Slices can be taken at each x_i, and treated as above to produce a complete sampling of both surfaces. We thus obtain a set of data representing the error of the Sketch Model estimate at each grid point. The error at a single point is given by:

$$\text{Error}_{i,j} = F(x_i, y_j) - G(x_i, y_j) \tag{6}$$

where

i & j = parameters of the sampling grid ($i,j = 1,17$)

$F(x_i,y_j)$ = altitude of the true composite function at the ij^{th} grid point

$G(x_i,y_j)$ = altitude of the Sketch Model estimate at the ij^{th} grid point

The accuracy measure is a combination of the errors at the individual grid points. It is formed by summing the squares of the error as defined by Eq. (6) and normalizing by the true composite function. This is shown below in Eq. (7).

$$\text{Accuracy Measure} = \frac{\sum_{i=1}^{17} \sum_{j=1}^{17} \left[F(x_i, y_j) - G(x_i, y_j)\right]^2}{\sum_{i=1}^{17} \sum_{j=1}^{17} F(x_i, y_j)^2} \tag{7}$$

This measure is similar to a relative mean square error. The measure is insensitive to the sign of the error; it also emphasizes large errors, due to the fact that they are squared.

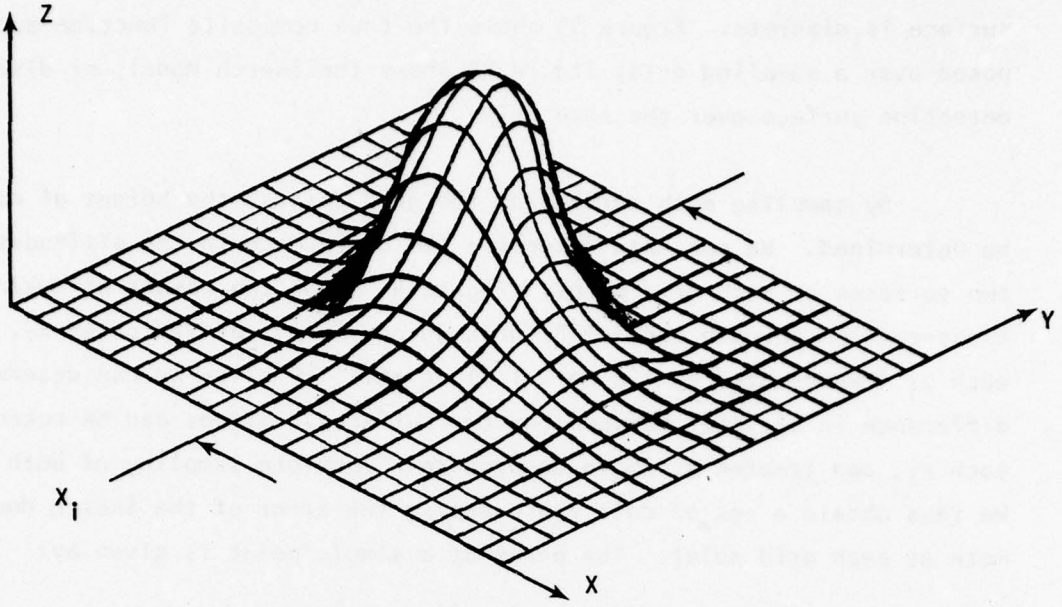


Figure 39. Three-Dimensional Representation of True Detection Surface.

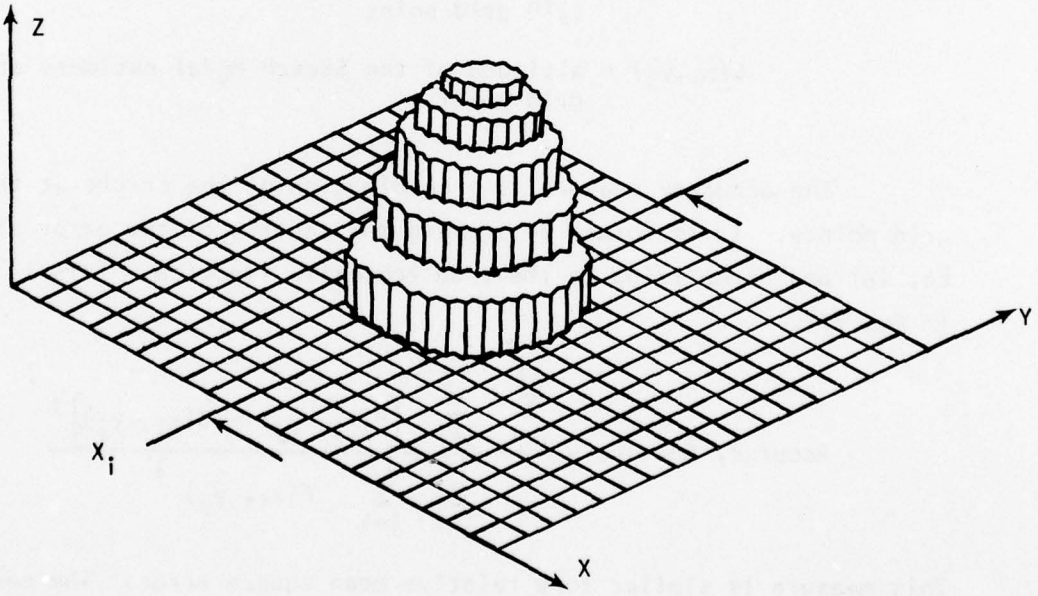


Figure 40. Three-Dimensional Representation of Sketch Model Surface.

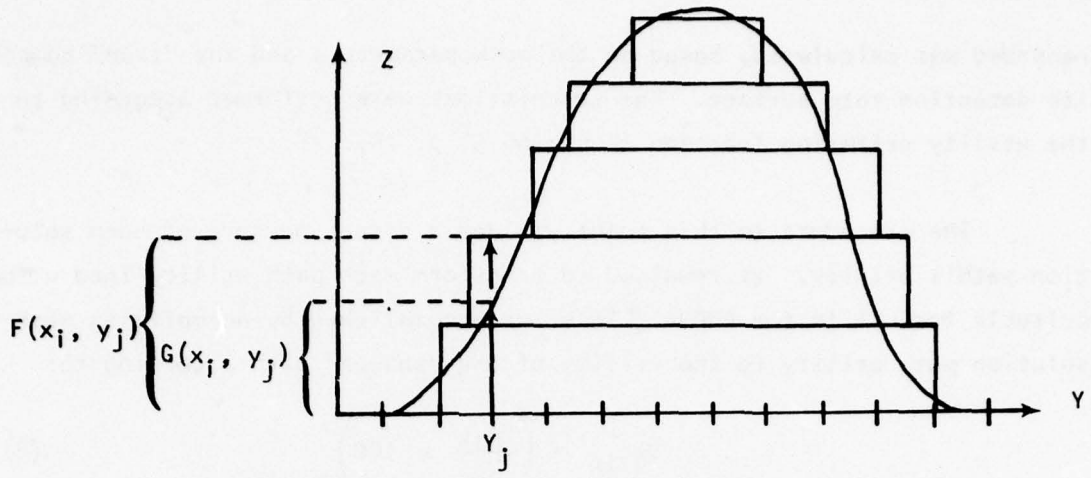


Figure 41. Superimposed Cross Sections of True and Sketch Model Detection Surfaces.

b. Path Utility. Recall that under the system concepts, an optimal strike path solution was generated for each problem either by each subject operator or by the computer. Path utility was chosen as the quantitative performance measure; in order to arrive at values of the solution path utilities usable in the ANOVA, data were collected and processed in the following manner.

First, the strike path parameters for the optimal path for each problem run under each system concept were recorded as raw data. The parameters, which fully specified each path solution, were

1. The x-y coordinates of the strike path launch and target points,
2. The x-y coordinates of the strike path "way points" (i.e., the grid points defining each path leg), and
3. The speed of each leg of the path.

For each problem run under System Concept A, the strike path solution was taken to be the last path that the operator completed. Under System Concept B, the data were collected for the operator's best path solution to a given problem. System Concepts C and D each generated one path solution per subject per problem.

Second, these strike path parameter data were used as input for a separate data reduction routine. In this routine the utility for each path

recorded was calculated, based on the path parameters and the "true" composite detection rate surface. The calculations were performed according to the utility criterion function (Equation 5, p. 24).

The procedure to this point yielded a direct measure of each solution path's utility. It remained to transform each path utility into a form suitable for use in the ANOVA. This was accomplished by normalizing each solution path utility to the utility of the "answer" path according to:

$$U_{N_{ijk}} = \left(\frac{U_{ijk}}{U_o} \times 100\right) \tag{8}$$

where

$U_{N_{ijk}}$ = normalized path utility of the i^{th} problem of the j^{th} subject of the k^{th} system concept

U_{ijk} = "direct" path utility

U_o = optimal (or "answer") path utility

4. Scenario Variables

The elements that defined a given problem within the experimental set were the adaptation of the ONRODA airstrike scenario area map (Section II.A), the independent variable of problem difficulty (i.e., intelligence reliability level), and three additional variables, not experimentally controlled, which were: (1) sensor locations, (2) reported sensor size, and (3) strike path start point.

a. Sensor Locations. Four sensor locations were defined for each trial: one location was fixed on ONRODA Island and the remaining three were located in the ocean area west of ONRODA, between the island and the task force. Accordingly, a pseudo-randomization scheme was devised to establish candidate configurations for the locations of each of the three ocean-platform-mounted sensors. First, each sensor was assigned to a specific region of the "ocean" area. Figure 42 depicts the size and location of each region with respect to the boundaries of our scenario adaptation. The figure shows, for instance, that one of the three ocean-located sensors will always be located in the 200 x 200 n.m.^{2} region west and north of ONRODA Island. It was assumed that the possible locations of a sensor

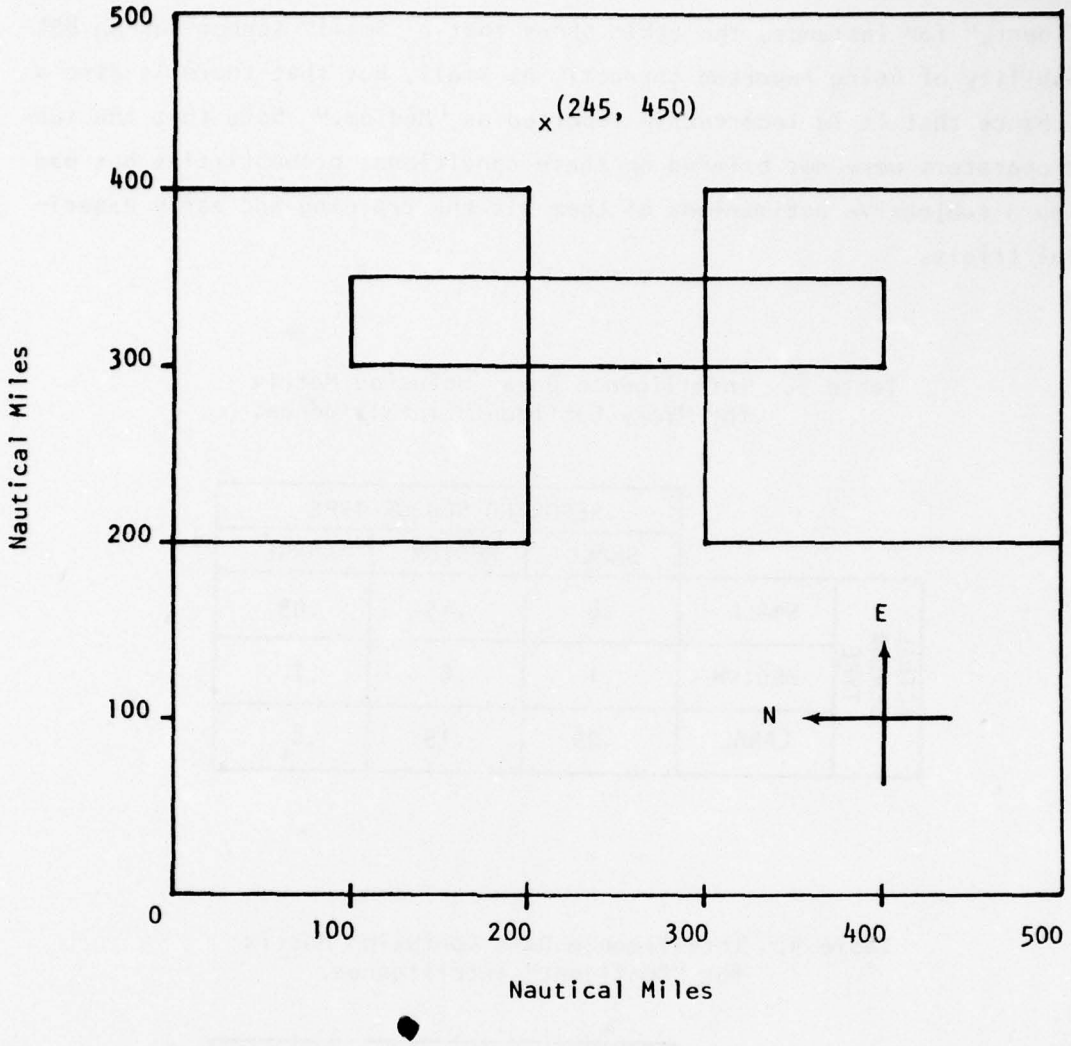


Figure 42. Sensor Location Randomization Boundaries.

within its region belonged to a uniform distribution. Accordingly, a random x-y pair was independently generated for each region, then each pair was normalized to the "window" of its region to give the coordinates of the sensor's location in that region.

b. Reported Sensor Size. For each trial, a single intelligence reliability level and the true size of each sensor were pseudo-randomly selected. Determining what sensor sizes to report to the operator was done by entering a conditional probability table (Tables 3 - 5) set up to correspond to the chosen reliability level. Table 3 provides an example of how

AD-A039 853 INTEGRATED SCIENCES CORP SANTA MONICA CALIF F/G 12/2
EXPERIMENTAL INVESTIGATION OF SKETCH MODEL ACCURACY AND USEFULN--ETC(U)
MAY 77 G W IRVING, J J HORINEK, P Y CHAN N00014-75-C-0811
UNCLASSIFIED ISC-215-3 NL

2 OF 2
AD
A039853

END

DATE
FILMED
6-77

this worked. When the intelligence reliability level is reported as "Very Confident," for instance, the table shows that a "Small" sensor has an 80% probability of being reported correctly as small, but that there is also a 15% chance that it is incorrectly reported as "Medium." Note that the subject operators were not briefed on these conditional probabilities but had to learn subjective estimations of them via the training and early experimental trials.

Table 3. Intelligence Data Confusion Matrix for "Very Confident" Intelligence.

		REPORTED SENSOR TYPE		
		SMALL	MEDIUM	LARGE
ACTUAL SENSOR TYPE	SMALL	.8	.15	.05
	MEDIUM	.1	.8	.1
	LARGE	.05	.15	.8

Table 4. Intelligence Data Confusion Matrix for "Confident" Intelligence.

		REPORTED SENSOR TYPE		
		SMALL	MEDIUM	LARGE
ACTUAL SENSOR TYPE	SMALL	.6	.25	.15
	MEDIUM	.2	.6	.2
	LARGE	.15	.25	.6

Table 5. Intelligence Data Confusion Matrix for "Not Confident" Intelligence.

		REPORTED SENSOR TYPE		
		SMALL	MEDIUM	LARGE
ACTUAL SENSOR TYPE	SMALL	.4	.35	.25
	MEDIUM	.3	.4	.3
	LARGE	.25	.35	.4

c. Strike Path Start Point. For each combination of four sensor locations (one sensor location from each of the three regions plus the "fixed" location) generated, the "true" composite detection rate contours were calculated according to the model (Section II.C.2) and printed out. The resulting set of detection rate contours was then examined to select those for which an optimum strike path was not necessarily obvious. Forty-eight combinations of sensor locations were chosen in this manner (reproduced in Appendix C). Finally a strike path start point was subjectively selected, specific to each set of contours. The 48 combinations of sensor locations, resulting "true" detection rate contours, and the selected start point were thus chosen in an attempt to assure that the problems would be non-trivial.

5. ANOVA Designs

a. Comparison of Operator Groups Under System Concept A. The purpose of this experiment was to determine if prior experience at Sketch Modeling biased operator performance on the same task done without Sketch Models. The design was a three-factor, fully replicated, randomized block factorial. The factors were:

1. Operator Groups--Two levels
   (Experienced and naive: four subjects each)
2. Problem Difficulty--Four levels
   (Denoted by intelligence reliability)
3. Replications--Two levels
   (First half of trials versus second half)

The observed variable was strike path utility. Data were collected from 48 problems performed by each of eight subjects. There were six problems per difficulty level per replication.

b. Comparison of Path Utilities Generated Under System Concepts A, B, and C. The purpose of this experiment was to investigate the usefulness of Sketch Models by comparing the effect of possible differences in system concepts on path utilities. The design was a four-factor, fully replicated, randomized block factorial. The factors were:

1. Operators--Four levels
   (Four U.C.L.A. undergraduates, trained in Sketch Modeling)
2. System Concepts--Three levels
   (System Concepts A, B, and C)
3. Problem Difficulty--Four levels
4. Replications--Two levels

The observed variable was path utility. Data were collected from 48 problems performed by each of the four subjects. There were six problems per difficulty level per replication.

c. Comparison of Variations in Errors Between Sketch Model Contours and True Contours. The purpose of this experiment was to determine if differences in the experimental factors were a significant source of Sketch Model error. The design was a three-factor, fully replicated, randomized block factorial. The factors were:

1. Operators--Four levels
   (Four U.C.L.A. undergraduates, trained in Sketch Modeling)
2. Problem Difficulty--Four levels
3. Replications--Two levels

The observed variable was the relative mean square error (Section IV.A.3.a) between the Sketch Model and true contours. Data were collected from 48 problems performed by each of the four subjects. There were six sets of Sketch Model contours drawn by each subject per difficulty level per replication.

B. TRAINING OF SUBJECTS

1. System Concept A

Subject operator training for System Concept A was conducted in three phases: orientation, introduction to the equipment, and training for system operation. Orientation was conducted lecture-style to the full group of subject operators. At this time they were introduced to the scenario problem of optimizing an air strike path through a detection field, and they were given explanations, without reference to the underlying analytical models, of the following aspects of the problem:

1. Single sensor detection rate model
2. Additive effect of sensor detection rate overlaps
3. Intelligence reliability
4. Path utility score components

The subjects were then familiarized with the use of the equipment necessary for system operation. Each received individual "hands-on" training until he was comfortable with equipment use.

The system operation training entailed each operator completing a set of training trials, using the same equipment and following the same experimental procedure as for the data-gathering sessions. System operation during the training trials was the same as that described in Section III.A, but with two important exceptions. One, of course, was that training problems were different from those to be run under test conditions. Secondly, two forms of feedback, not implemented in the subsequent data gathering trials, were supplied to each subject during each of his training trials. Feedback was provided on (1) the operator's choice of the 10% detection range for each sensor, and (2) the utility of each strike path he completed.

The first feedback feature was provided so that the operator could develop a feel for the relationship between reported sensor types and the specified reliability of intelligence information. When the operator finalized his choice of sensor templates (as in Section III.A), which were represented as broken red circles on the display, an additional set of sensor templates, coded in green, was automatically displayed. Each green template represented the "true" 10% detection rate level of its associated sensor.

Figure 43 depicts an example of the display format for this feature. The figure is simplified in that it shows templates for only two of the sensors.

Subjects were instructed to draw their strike paths on the basis of the true (green-coded) sensor templates. When each path was finished, its utility was displayed to the subject. Each utility shown was based on the true composite detection function; subjects however were not told this, nor were they shown a complete set of the true detection rate contours.

The performance data collected from the training trials was used to determine whether any subject required additional training before undertaking data trials. All subjects, however, performed well on the training trials, so that further training was not conducted.

2. System Concept B

Training for System Concept B operation differed from that for Concept A in the following ways. First, subjects received more thorough instruction on the details of the single-sensor detection model, including the "volcano" effect and its impact on the configuration of a composite detection field. Second, they received extensive paper-and-pencil training in Sketch Modeling. The task was analogous to the Sketch Modeling they were to do on the display screen. Each subject received a set of problems, each consisting of four "sensor" locations and types. Using transparent sensor coverage templates, a subject investigated the areas of sensor coverage overlap. Removing the transparencies, he sketched contours representing the 90, 70, 50, 30, and 10% altitudes of the composite detection function. He was then given the "answer" contours on transparencies to compare with his own. When the Sketch Model training was complete, training trials were begun on the display. At this point, system operation was identical to that described in Section III.B, except that different problems were run.

C. EXPERIMENTAL PROCEDURE

The experimental procedure described below pertains to those data-gathering sessions that involved subject operators, i.e., those under System Concepts A and B.

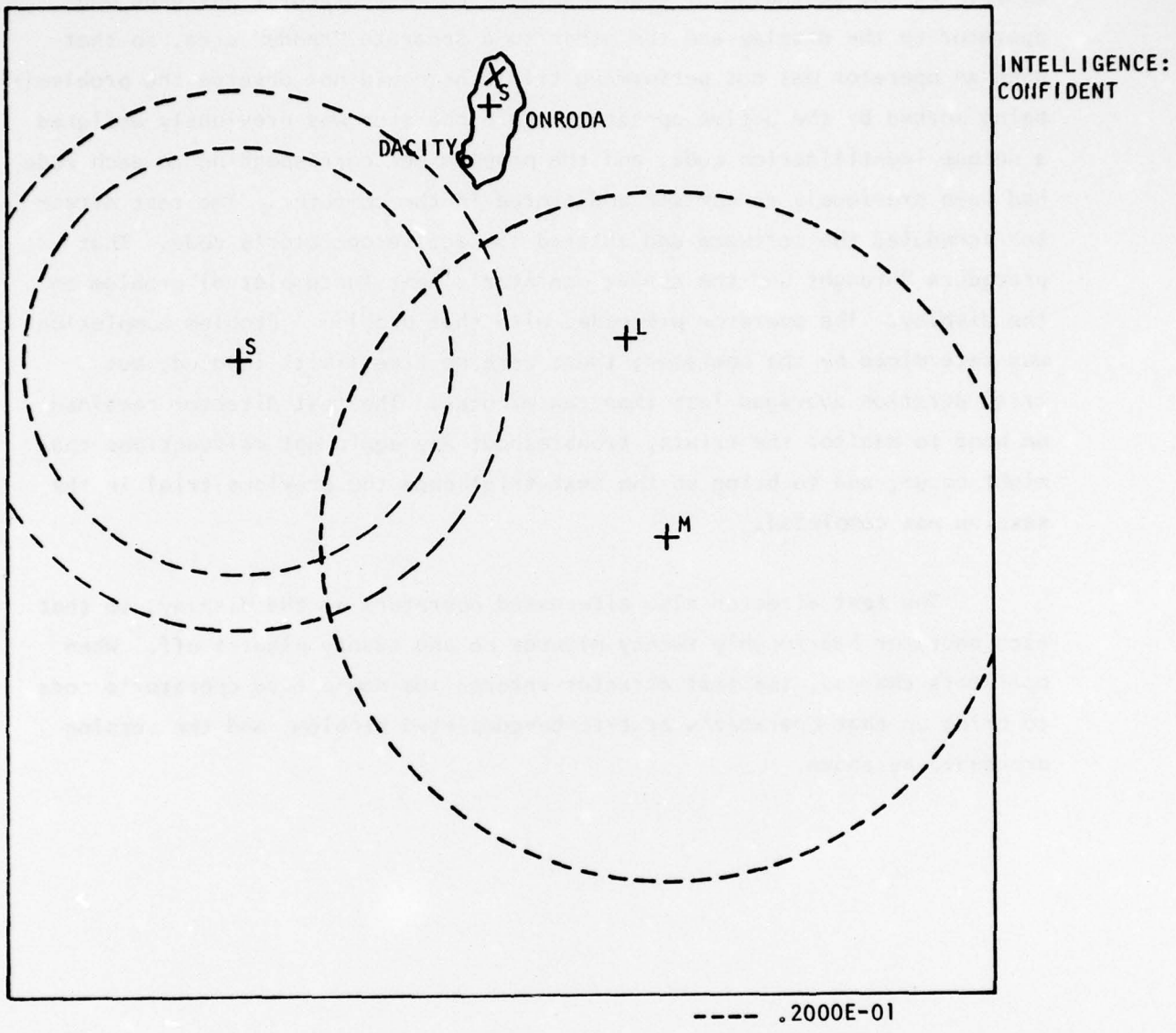


Figure 43. Display with Sensor Template Feedback, System Concept A (Training).

Each data gathering session was scheduled for two hours. The experimental team for each session included the test director and two subject operators. At the beginning of each session, the test director assigned one operator to the display and the other to a separate "ready" area, so that when an operator was not performing trials he could not observe the problem(s) being worked by the active operator. Each operator was previously assigned a unique identification code, and the problem set corresponding to each code had been previously randomized and stored in the computer. The test director scheduled the software and entered the active operator's code. That procedure "brought up" the active operator's next (uncompleted) problem on the display. The operator proceeded with that problem. Problem completion was determined by the operator; there were no time limits imposed, but trial duration averaged less than ten minutes. The test director remained on hand to monitor the trials, troubleshoot any equipment malfunctions that might occur, and to bring up the next trial once the previous trial in the session was completed.

The test director also alternated operators at the display, so that each operator had roughly twenty minutes on and twenty minutes off. When operators changed, the test director entered the now-active operator's code to bring up that operator's next-to-be-completed problem, and the session proceeded as above.

V. RESULTS

The analyses investigating the usefulness of Sketch Models hinged on first determining whether prior experience with Sketch Models biased subsequent operator performance on the same task done without the Sketch Model feature. The results (Section A, below) indicated that there was no such effect. This result meant that the usefulness of Sketch Models could be investigated by first comparing System Concepts A, B, and C and then comparing these to Concept D (Section B). Two additional analyses were performed (Section C) to investigate Sketch Model accuracy: (1) analysis of variance was used to identify sources of variation from among the experimental factors, and (2) percent volume errors were analyzed to assess Sketch Model accuracy and to identify sources of error inherent in the method itself.

A. ANALYSIS OF OPERATOR GROUPS UNDER SYSTEM CONCEPT A

A three-way analysis of variance was performed on the path utility data generated by the two operator groups, one naive and the other experienced with Sketch Models. After pooling procedures (Reference 6) were applied, the ANOVA results were as shown in Table 6. The critical result of this analysis was that the naive group performed significantly better than the experienced group. This result allowed data generated by the experienced subjects under System Concept A to be included in the following analysis (Section B). This was desired since Concept A, by not incorporating the Sketch Model aid, was intended to represent current operational practices.

B. ANALYSIS OF VARIATIONS IN PATH UTILITIES ACROSS SYSTEM CONCEPTS

This analysis was intended to investigate the usefulness of Sketch Models. Although the number of observations was large, the number of subjects was small, so that the results obtained were of limited use for this aspect of the investigation. The analysis began with a four-way analysis of variance performed on the utility data generated by the experienced subjects for each of System Concepts A, B, and C. The results (after pooling) are displayed in Table 7.

Table 6 . Analysis of Variance for Comparing Groups Under Concept A (After Pooling).

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Square	F_{obs}
Groups	1	7,134	7,134	22.25*
Problem Difficulties	3	1,949	649.7	2.03
Groups x Problems	3	1,190	396.7	1.24
Replications	1	10	10	0.03
Error	375	120,223	320.6	
Totals	383	130,500		

*$\alpha = 0.10$

Table 7 . Analysis of Variance for Comparing Concepts A, B, and C (After Pooling).

Source of Variation		Degrees of Freedom	Sum of Squares	Mean Square	F_{obs}
Concepts	(C)	2	1,427	713.5	2.99*
Problem Difficulties	(P)	3	1,659	553	2.31*
C x P		6	2,001	333.5	1.40
Operators	(O)	3	28,220	9,406.7	39.4*
C x O		6	6,863	1,143.8	2.94*
P x O		9	3,251	361.2	1.51
C x P x O		18	3,553	197.4	0.51
Replications	(R)	1	13	13	0.05
C x O x R		6	2,335	389.2	1.63
P x O x R		9	3,060	340	1.42
C x P x O x R		18	6,935	385.3	1.61*
Error		494	118,054	239	
Totals		575	177,400		

*$\alpha = 0.10$

Since the effect of the system concepts was significant, contrasts between pairs of system concepts were calculated and tested by Scheffé's method. Concepts A and C were found to be significantly different, but the other pairs (A and B, B and C) were not.

Data from Concept D were used as follows. Since, of the three concepts (A, B, and C), Concept C yielded the highest total utility, it was possible to compare D with the other concepts by simply comparing it with C. This was done by applying the Kolmogorov-Smirnov test to cumulative frequency distributions of utility values from Concepts C and D. This test showed that Concept D was significantly better. An additional comparison of the four system concepts was made by dividing the total utilities achieved under Concepts A, B, and C by the total utility under Concept D to arrive at a percentage of optimum score. The results were:

Concept	Percent of D's Utility
A	89.3%
B	92.1%
C	92.9%

The overall results of comparing the system concepts were that, numerically, Concept A < B < C < D, with significant differences observed only between A and C and between D and the other three. This, in effect, indicates that for our scenario it takes the combination of computer-optimized path solutions and subjects' Sketch Models (Concept C) to achieve significantly better paths than were gotten without aids (Concept A); Sketch Models alone produce no significant improvement in path utilities.

The lack of significant difference between System Concepts A and B was surprising, but an investigation of a higher-ordered effect provided a clue to a probable underlying cause. Figure 44 shows the interaction between operators and system concepts. Tests indicated that the performance difference within Concepts A and B was mainly due to Operator 1 and that Operator 1 was the only one whose performance was significantly improved by Concept C (dynamic programming optimization). This, combined with the fact (see Figure 44) that Operator 1 was the worst subject under all three concepts, leads to the suspicion that the experimental problems were not difficult enough to provide a valid test of the usefulness of Sketch Models.

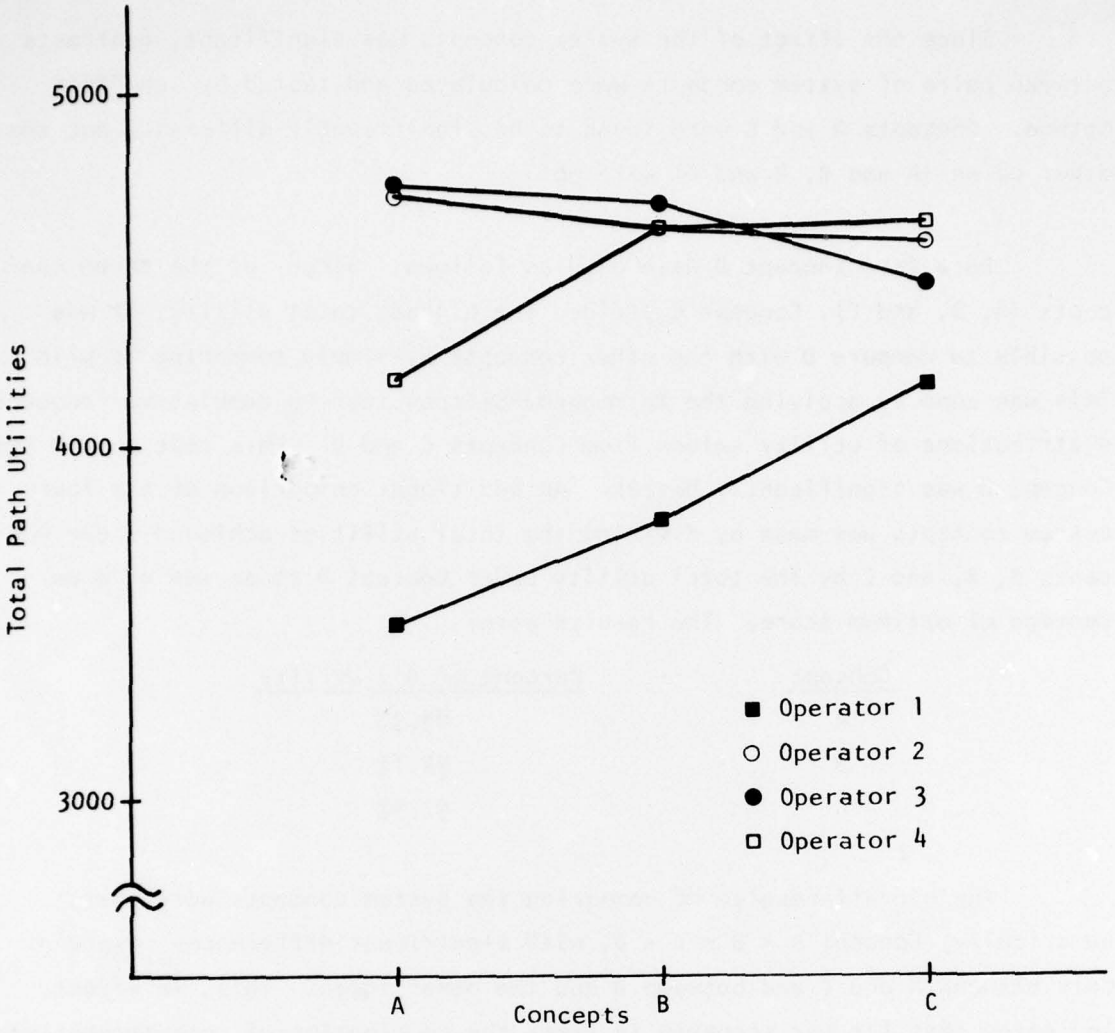


Figure 44. Plot of the Interaction Between Operators and System Concepts. (The lines do not indicate that the data are sampled from a continuum but are provided to aid the reader in perceiving the statistical interaction between the factors.)

C. ANALYSIS OF ACCURACY OF SKETCH MODEL CONTOURS

1. Analysis of Variations in Accuracy

The operator-drawn contours from Concept B were compared with the true contours from Concept D by a three-way analysis of variance of the relative mean square error. Table 8 shows that the significant sources of variation in Sketch Model accuracy were the main effects of operators and replications, and the interaction of problem difficulties with replications.

Table 8. Analysis of Variance for Contour Sketching Accuracy Data (After Pooling).

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Square	F_{obs}
Operators	3	.5210	.1737	16.62*
Problem Difficulties	3	.1342	.0447	1.84
Operators x Problems	9	.1014	.01127	1.08
Replications	1	.03827	.03827	3.67*
Problems x Replications	3	.07283	.02428	2.3*
Error	172	1.7980	.01045	
Totals	191			

*$\alpha = 0.10$

By applying Scheffé's method it was found that the variation due to operators was attributable to Operator 1, who was previously found to be the cause of the significance of the operator effect on path utilities. Of greater interest is the problem difficulties x replication interaction, plotted in Figure 45. The figure clearly shows that the significance of this interaction was due to the subjects getting substantially better at producing Sketch Models for the more difficult problems (Problems 1, 2, and 3 in the figure) in the second replication. Furthermore, the figure illustrates (and Scheffé's method confirms) that for the second replication there was no significant difference across problem difficulty levels. In other words, the subjects not only improved on the harder problems (i.e., those with greater uncertainty), but also did about as well with those as with the easiest problems, where there was no uncertainty.

2. Analysis of the Percent Volume Error of Sketch Models

One measure of the similarity between a true detection rate function and its corresponding Sketch Model is the volume of the difference between their two surfaces. If this absolute error is normalized by dividing by the volume under the true detection rate function, the resulting measure can be called "percent volume error."

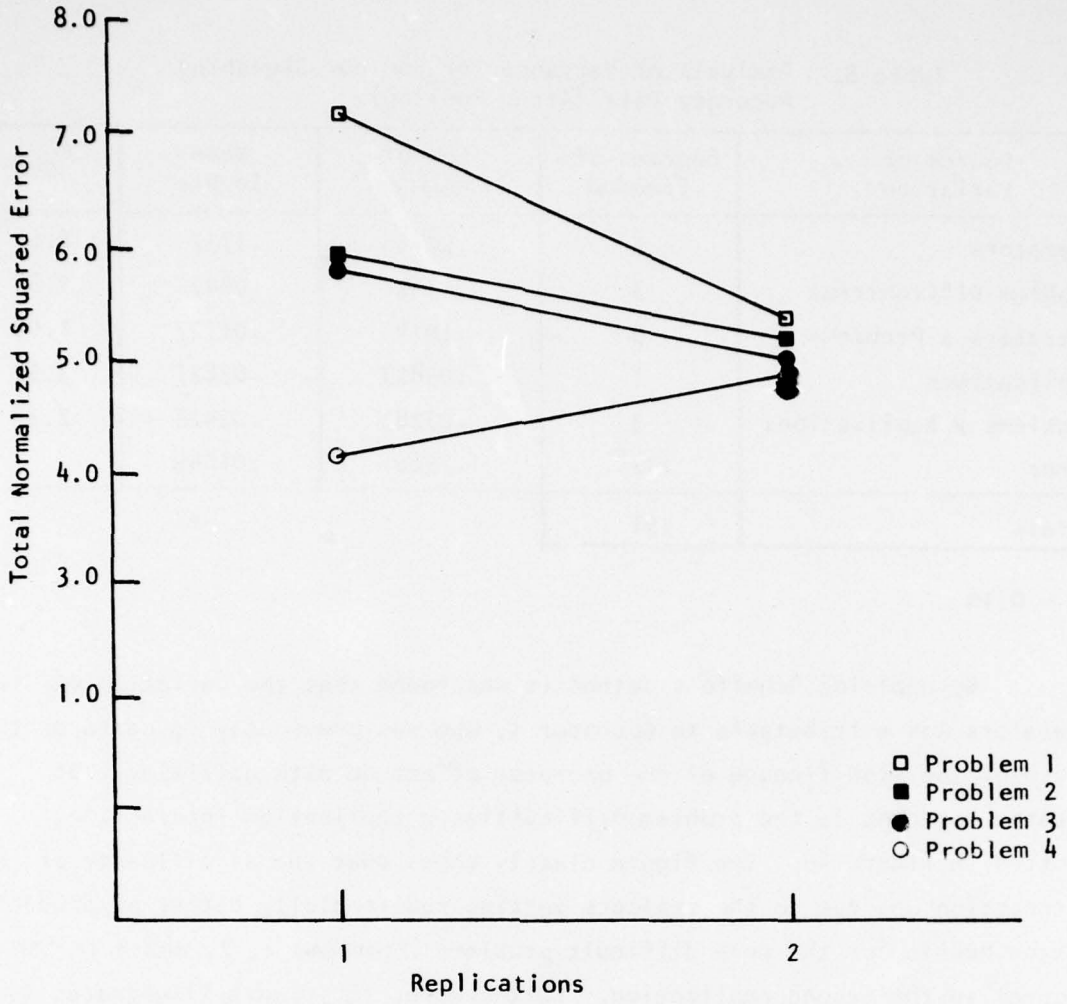


Figure 45. Plot of the Problem Difficulty x Replication Interaction for the Contour Sketching Accuracy Data.

The percent volume error was calculated for each of the 192 Sketch Models developed by the subjects. Figure 46 shows the frequency-of-occurrence histogram of the percent volume error measures for all of the trials. It is unimodal but skewed to the low error side and therefore *not* Gaussian. The bulk of percent volume error values fall between 20 and 50 percent. A median value of the histograms would fall around 36 percent volume error. These percent volume error values are difficult to view from a pragmatic perspective without a meaningful frame of reference. Are they good? Did the subjects do well? Before these questions can be answered others must be answered. How good is good? What percent volume errors would have

resulted if the subjects had been completely inept, i.e., totally ignored the intent of the task?

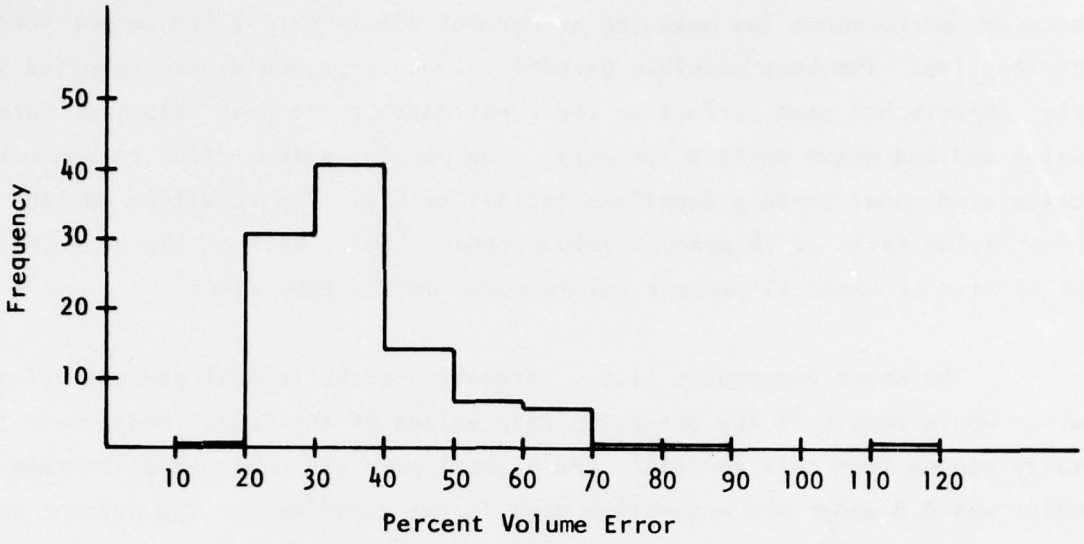


Figure 46. Frequency of Percent Volume Error for All Problem Difficulty Levels.

A good handle can be gotten on the answers to these questions by identifying and isolating the various sources of the percent volume error. The three primary sources of error are:

1. Detection Rate Peak Value Estimate. As described in Section III.B, one step in fully specifying a Sketch Model was to use scalar subjective judgment to estimate the detection rate value of the maximum point (peak) of the true detection rate surface. Since the subjects were not perfect in this task, this was the first source of percent volume error.

2. Contour Drawing. Even if the subjects had been perfect in their estimation of the peak detection rate value, there still would have been percent volume error due to their not drawing the contours perfectly.

3. Discrete versus Continuous. The fact that the Sketch Model is discrete in the detection rate (z) dimension and is being compared to a continuous detection rate function is a source of error. Therefore, even if the subjects had been perfect in both their estimate of the peak detection rate value and their contour drawing, there still would have been some finite percent volume error.

The percent volume error values measured from the Sketch Models thus fall on a metric with the best (highest) and worst (lowest) possible values defining the scale. These values must be determined before the Sketch Model accuracy performance (as measured by percent volume error) can be put into perspective. The best possible percent volume error would have resulted if the subjects had been perfect on their estimate of the peak detection rate value and had drawn perfect contours. The percent volume error values were calculated under these assumptions for all trials. The resulting median error value falls at 14 percent volume error. Thus, half of the time the subjects have at least 14 percent volume error before they start.

The worst reasonable (i.e., largest, non-capricious) percent volume error would result if the detection rate values of the Sketch Model were totally random from cell to cell. The highest possible peak detection rate value was 0.8 under the assumption used in the experiment. The percent volume error was calculated for each trial with the detection rate value from the Sketch Model being substituted by a random number drawn from a uniform density function between 0 and 0.8. The median percent volume error under these conditions is 142 percent.

Figure 47 gives a thermometer scale (on the left) showing the best and the worst possible percent volume errors (median values). Between them is the median value of the percent volume error for the Sketch Models averaged over all problem difficulty levels. Also shown is the median value of the percent volume error for the "perfect information" problem difficulty level. It is obvious from Figure 47 that the subjects performed quite well on the Sketch Modeling task. Note that on the accuracy scale (on the right), the subjects' Sketch Models scored 84 over all problem difficulty levels.

Also of interest were any systematic biases operating in the Sketch Model behavior. Were certain detection rate values systematically estimated low or high? The points from each contour of each trial were aggregated by true detection rate interval. The mean and standard deviation were determined for all the Sketch Model detection rate estimate points falling in each interval. The results are plotted in Figure 48. The mean values of Sketch Model estimates are shown as dots while the dispersion ($\pm 1\sigma$) of each estimate is shown as a vertical line extending from the dot.

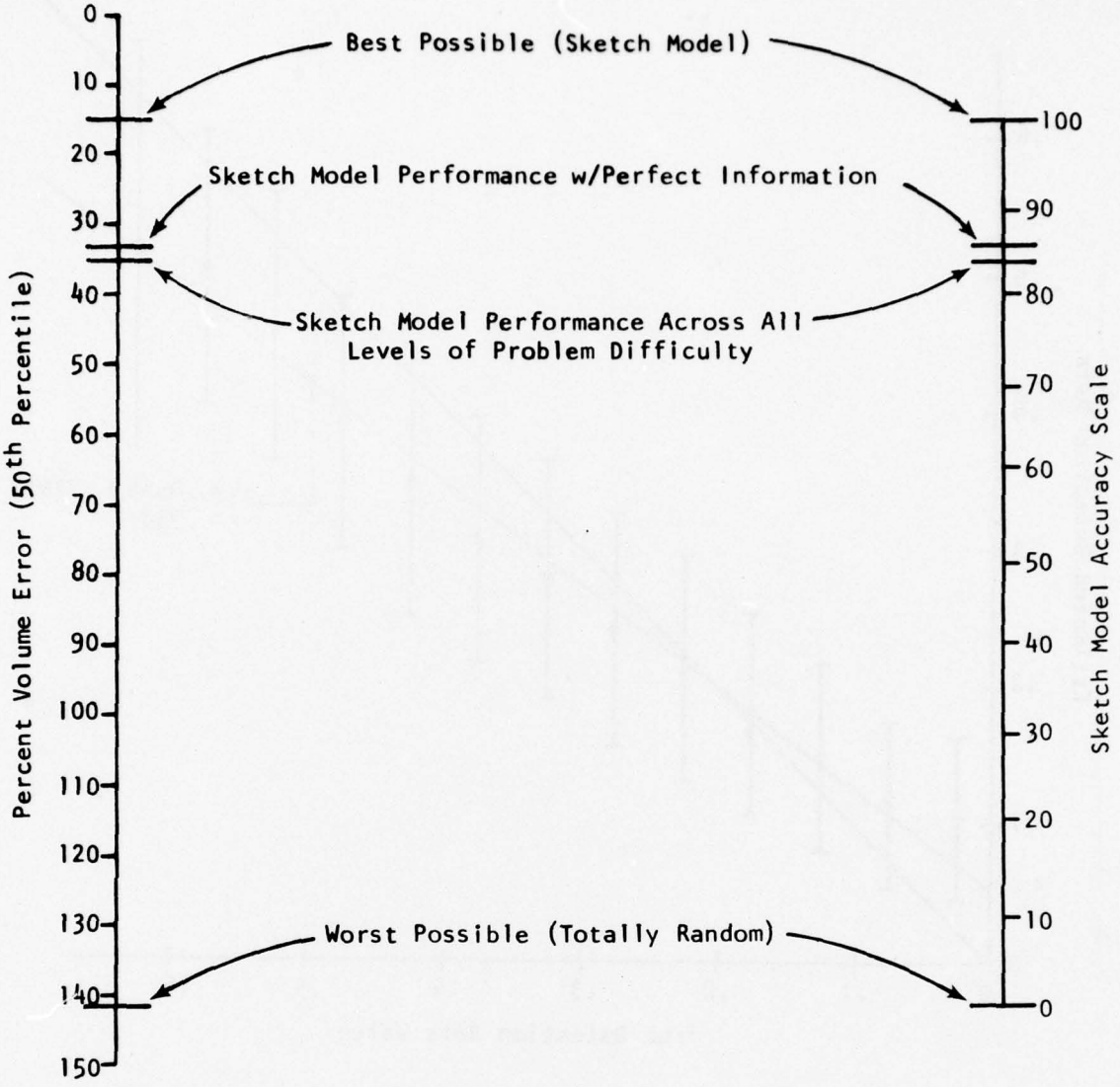


Figure 47. Scales for Percent Volume Error (50th Percentile).

A number of features are immediately obvious. First, the mean value dots fall almost perfectly on a straight line (correlation coefficient of .993) indicating that the contours were drawn in a very consistent manner. Secondly, the line of mean value dots shows the classical central tendency effects of scalar subjective judgment. The central tendency effect results from high true detection rate values being consistently underestimated while low values are overestimated. Finally, the small dispersion ($\pm 1\sigma$) range, relative to the total allowable scale, is another indication of the good estimation performance achieved by the subjects using the Sketch Model technique.

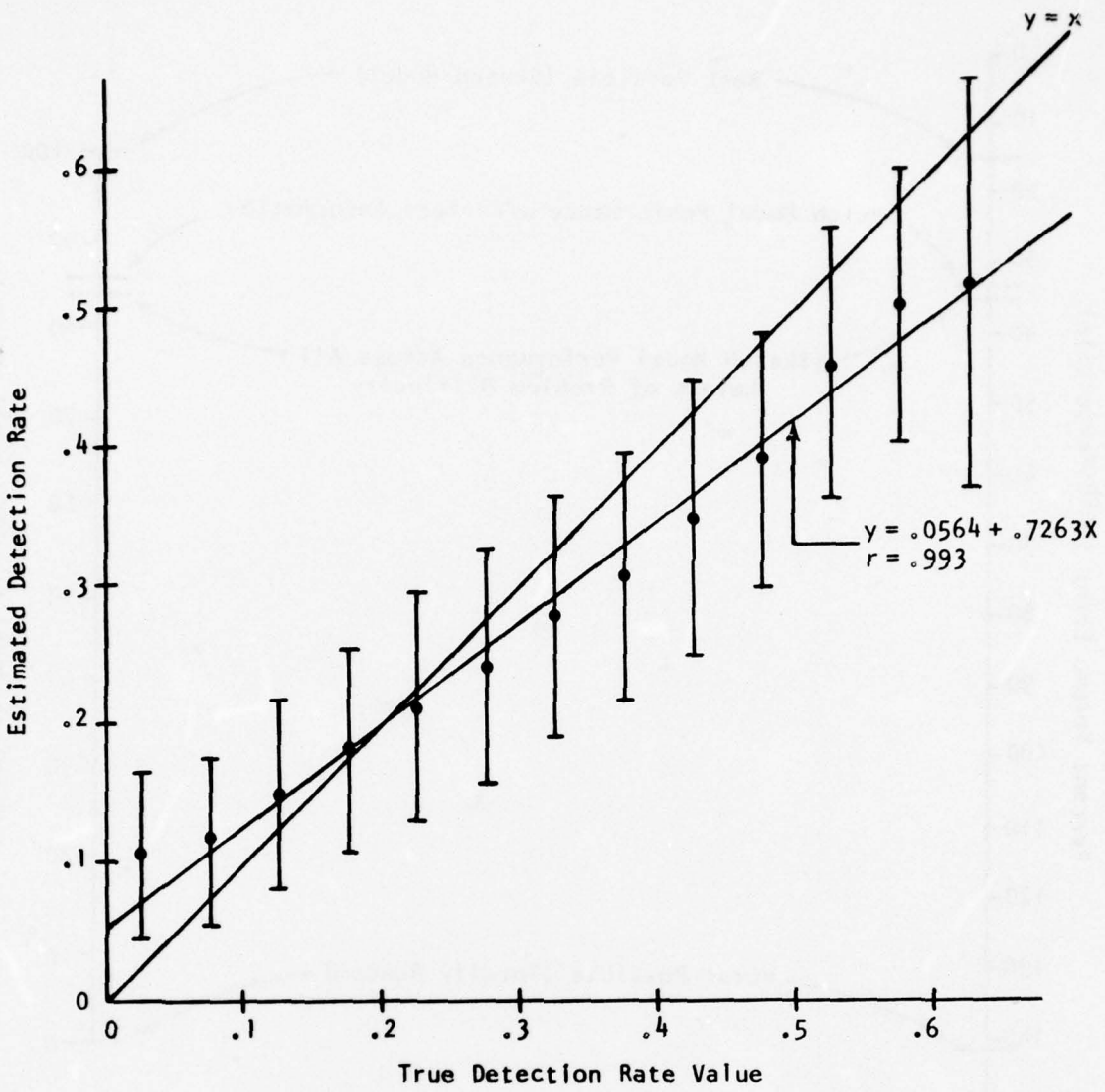


Figure 48. Dispersion of Sketch Model Estimates with Respect to True Detection Rate Values.

The central tendency effect could have resulted from either a central tendency effect on the scalar subjective judgment estimation of the peak detection rate value or by how the contours were drawn or both. A plot of the mean scalar subjective judgment values for each true peak detection rate interval is shown in Figure 49. Once again the mean value dots fall almost on a straight line (correlation coefficient of .972) and there is a general central tendency effect. The central tendency effect is as expected since these are results from a scalar subjective judgment procedure.

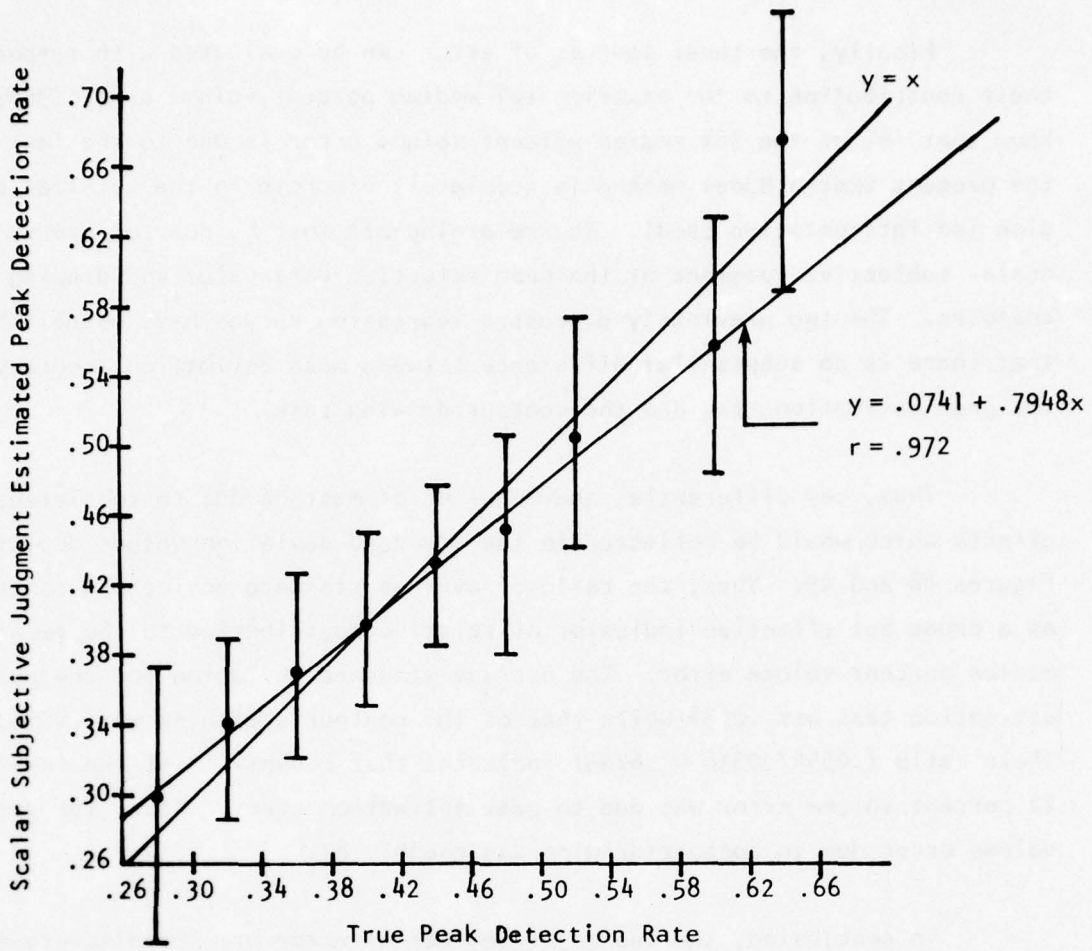


Figure 49. Mean and Dispersion of Scalar Subjective Judgment Estimates of Peak Detection Rates.

The crucial fact concerning Figure 49 is that the result of a linear regression fit is very close to the linear regression fit to the data in Figure 48. In other words, the central tendency effect seen in the Sketch Models' detection rate estimates is due almost completely to a central tendency bias originating in the scalar subjective judgment process used to estimate the peak value of the composite detection rate function. It is also clear from examining Figures 48 and 49 that most of the errors result from the scalar subjective judgment process. This means that a very small amount of additional error was introduced by the contour drawing portion of the Sketch Model development. Thus, the subjects did an excellent job of drawing the Sketch Model contours accurately.

Finally, the three sources of error can be evaluated with respect to their contribution to the experimental median percent volume error (36%). We know that 14% of the 36% median percent volume error is due to the fact that the present Sketch Model method is completely discrete in the vertical dimension (no interpolation used). The remaining 22% must be due to errors in scalar subjective judgment of the peak detection rate value and drawing of the contours. The two previously discussed regression curves have established that there is no substantial difference between mean estimation accuracy of the peak estimation task and the contour-drawing task.

Thus, any differential source of error must be due to consistency effects which would be reflected in the standard deviation values depicted in Figures 48 and 49. Thus, the ratio of average standard deviations can be used as a crude but effective indicator of relative contribution to the remaining median percent volume error. The average standard deviation for the peak estimation task was .0594 while that of the contour sketching was .0936. Their ratio (.0594/.0936 = .6343) indicates that roughly 14 of the remaining 22 percent volume error was due to peak estimation error. Thus, the percent volume error due to contour-drawing was roughly 8%.

In conclusion, the two major sources of error are the discreteness of Sketch Models in the vertical dimension and the peak estimation task, and they are roughly equal (14%). The third identifiable source, the contour-drawing task, is substantially smaller than the other two (8%). This is important since the other two sources of error are much more amenable to improvement by transformation and interpolation procedures.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. UNDERLYING CAUSES

The results of Section V and any interpretations thereof painfully point out the complexities and paradoxes of aided decision making research. If the purpose of aided decision making research is to develop techniques to help make better decisions, the research effort is only meaningful when applied towards problems which presently defy solution or at least prevent acceptable good decisions. On the other hand, if one is to do valid research, one should test operator performance on problems to which a true answer can be found. Otherwise the researcher is hard pressed to measure decision performance and thus is in a weak position to determine whether the decision aid he evaluated significantly improved decision performance or not. Therein lies the key paradox: if the researcher evaluates a proposed decision aid on a real problem he can't validate performance improvement, and if he evaluates the aid on a laboratory problem with a known true answer he can't completely know how the results would extrapolate to the real world.

As in the case with other paradoxes, the researcher forges ahead, seeking to make do with compromises. In this case the researcher must select a problem of sufficient complexity to be representative of reality while not consuming all his resources in an effort to solve for the true answer. A substantial attempt was made in this study to develop problems which would be complex enough to truly challenge the subjects while not requiring an inordinate amount of effort on the part of the researcher to solve for the correct answer.

It was originally desired, for instance, to include aircraft altitude in the air strike path solution vector. When it was determined that this would increase the dimensionality of the problem such that the time required for the dynamic programming algorithm to solve for the "best" air strike path ("true" answer) was unacceptable, the aircraft altitude dimension was sacrificed.

One of the original purposes in selecting the volcano model for the single sensor detection rate function was that high detection rate derivatives occurring between the rim and hole of the volcano would produce complex interactions when combined with other sensors, resulting in irregular composite surfaces on which to test the Sketch Model technique. While producing a detection rate surface of sufficient complexity, the volcano model also leads to the disconcerting feature that many of the "true" path solutions produced by the dynamic programming algorithm fly directly over at least one sensor, thus calling the realism into question. This problem could have been solved by using simple cone-shaped models of single sensor detection rate functions, but then the resulting composite multi-sensor surfaces would have been too easy to warrant the use of Sketch Models.

The original sensor random placement algorithm led to the sensors being bunched in the middle of the CRT screen. This encouraged the trivial solution of always going around the sensors, circumventing the probability of detection problem. Consequently the problems were redefined by spreading the sensor placements so as to prevent a circumventing path from being the trivial solution. This forced portions of the composite detection rate surface contours off the rectangular screen coordinates. Thus, an additional software effort was required in order for the subjects to be able to "sketch" a contour right up to and along the edge of the addressable screen coordinate frame.

The air strike path utility function was specifically designed with high derivatives (sharp "elbows") in order to increase sensitivity to small changes in an air strike path. As a result of the interaction between the composite detection rate surfaces, the fuel consumption model, and the shape of the utility function, most of the problems could be solved with very high (>85) utility. Basically all the answers were so high that the subjects spent all their time refining their answers in a utility function region to one side of the elbows, a region which was essentially linear. This meant that all of the effort expended to develop a meaningful yet highly nonlinear utility function was wasted.

Further time was spent to investigate the sensitivity of the dynamic programming solution to the location of the air strike path launch point.

Particular attention was paid to situations in which a small change in the launch point resulted in a substantial change in the solution (e.g., from a slow, circumventing path to a fast, "up the middle" and straight-in path). With a feel for where the ambiguity regions of starting points lay, each starting point of each experimental problem was specifically selected to maximize ambiguity and thus problem difficulty. The result, however, appears to have been that too many of the solution paths were straight-in and that this became the trivially obvious solution.

The fuel consumption rates were selected to be nonlinearly related with aircraft velocity in order to be realistic. This resulted in so high a fuel utility price being paid for the highest speed that it was never used once in any of the dynamic programming solutions. The subjects quickly caught on and never used it either. This had the effect of simplifying all the problems, once again defeating the goal of problem complexity.

All of these problems were potentially avoidable had there been time to explore the consequences of each decision. Designing the trial problems was a much more complex and highly dimensioned task than was solving those problems. This is always the case; the experimenter by definition, in trying to define appropriate problems, is struggling with a greater problem of higher ordered complexity than the ones on which he purports to investigate decision aids. Since schedule realities invariably demand that such problem design decisions be made in parallel, the risk of producing a set of problems either all-too-easy or all-too-difficult is ever present. Such was the case in this experiment with the result being that the problems were too easy. This is apparently the case in spite of all the concern for, and effort directed at, producing sufficient problem difficulty.

B. CONCLUSIONS

The statistical analysis alone is not sufficient for drawing conclusions, interpreting results, and identifying possible causal phenomena that explain the patterns of statistical results. A thorough understanding of the assumptions inherent in the operating characteristics of the models used in the scenario (e.g., detection rate model, utility function, fuel rate data, intelligence reliability model, etc.) is required. Many of these

features have been discussed in Section VI.A. Also required is a thorough familiarity with the subjects' training, their operational rules of thumb, and their level of understanding of the problem. The following conclusions and supportive interpretations have been made from such a background.

Conclusion 1: Only partial evidence is available to support the hypothesis that the Sketch Model technique aided path selection decision making as measured by air strike path utility, and this is likely due primarily to the fact that the air strike path selection problems were too easy.

Interpretation: In comparing path utility performance for System Concepts A, B, and C, the only significant difference was that C was better than A. Thus it was only when the Sketch Model was combined with the dynamic programming algorithm that it contributed to a performance improvement over System Concept A, the experimental representative for present methods.

The expected performance superiority of System Concept B over System Concept A failed to materialize. This may be the case either because Sketch Models do not (in the air strike scenario) help that much by themselves or because the air strike path selection problems were so easy that the solutions were even obvious without Sketch Models as a guide. Appendix C illustrates the true multi-sensor composite detection rate contours with the optimal air strike path as selected by the dynamic programming algorithm. It is evident that a considerable portion of the optimal paths consist of relatively straight lines between the starting and ending points with much of the path transited at medium aircraft speed. This solution archetype is something a subject could pick up on very quickly. Also, such a solution is relatively insensitive to Sketch Model accuracy since the use of the medium velocity overcomes most probability of detection penalties regardless of what detection rate regions the path passes through.

Consequently, it is highly likely that the ease with which the optimal path solutions were identified by the subjects is an appropriate explanation for two unexpected results. First, the lack of significant difference between System Concepts A and B suffers a loss of credibility because of path problem simplicity. It should not be interpreted as a strong indication that

the Sketch Model doesn't really help in an air strike path application. Secondly, the lack of significant difference between System Concepts B and C should not be interpreted as supporting the conclusion that subjects can do as well as dynamic programming at air strike path selection. The relatively easy problems used in the experiment were not a true test.

Actually, simplicity of a problem is a relative matter. It is a combination of problem difficulty and the decision power of the person tackling the problem. This match-up resulted in easy problems for three subjects. However, these three subjects (2, 3, 4) were significantly superior to all performances measured from Subject 1. Thus, the problems were difficult for Subject 1, and those results provide a truer test of performance improvement attributable to the Sketch Model. In Subject 1's data alone, System Concept C is significantly superior to System Concept B, which is significantly superior to System Concept A.

The interpretation is that for persons of lesser decision making capability, the Sketch Model significantly aids performance as does the dynamic programming algorithm. This in conjunction with the superiority of System Concept C over System Concept A for all subjects can be interpreted as a partial indication that Sketch Models do aid air strike path selection decision performance. This in turn is a partial indication that humans can use the Sketch Model technique well enough on irregular functions (multi-modal, unsymmetric, etc.) that it is useful as a decision aid.

Conclusion 2: There is very strong indication that the subjects were able to produce accurate (as measured by percent volume error) Sketch Models for the highly irregular (multi-modal and unsymmetric) multi-sensor detection rate surface.

Interpretation: On a percent volume error scale normalized by the best and worst possible values to 100 points, the median value of the subjects' Sketch Model percent volume error scored 84 points. A plot of the mean and standard deviations of the Sketch Models' detection rate estimates versus intervals of the true detection rate values (Figure 48) provides additional support to Conclusion 2. The mean points fell almost exactly on

a straight line (correlation coefficient of .993) indicating a very systematic process. The straight line was only slightly tilted ($\theta = 36°$) from the perfect estimate (45°) line indicating high accuracy with a slight central tendency bias. The standard deviations were all roughly equal regardless of the mean estimate value, indicating a consistent process.

Conclusion 3: The present Sketch Model procedure (described in Section III.B) can be significantly improved upon (with respect to percent volume error performance) by transformation and interpolation functions.

Interpretation: Of the three sources of percent volume error the largest two were discrete-versus-continuous mismatch and peak detection rate value estimate by scalar subjective judgment. Errors from these sources were much larger than errors resulting from the subjects drawing imperfect contours for their Sketch Models. Much can be done on the basic contour data by way of interpolation to create a smoother Sketch Model, less discrete in the vertical dimension. This would serve to greatly reduce the percent volume error resulting from comparing the discrete Sketch Model with the continuous true surface.

The central tendency bias on the scalar subjective judgment estimate of the detection rate peak value is a classical and very predictable one. This allows it to be corrected for within the Sketch Model procedure by a simple transformation that compensates for underestimation of large values and overestimation of small detection rate peak values.

Conclusion 4: There is evidence of Bayesian behavior taking place in the multi-sensor detection rate surface Sketch Model task.

Interpretation: There was a significant interaction between problem difficulty and replications with respect to the Sketch Model accuracy measure. This was shown to result from significant improvement in Sketch Model accuracy on the more difficult problems (intelligence reliability--not confident) between replication 1 and replication 2. This suggests a learning effect whereby the subjects learned how to adjust their Sketch Models to account for likely intelligence errors. In order to do this with any effectiveness

they must have had some appreciation of the *a priori* probabilities of intelligence error. In considering these *a priori* probabilities the subjects inserted some Bayesian aspect into their Sketch Model development decision making.

For the second replication, there was no significant difference in the accuracy measure between levels of problem difficulty. Thus, the subjects learn to completely neutralize any impact of intelligence reliability on Sketch Model accuracy. This was not the result of a central tendency effect where the subjects learn to produce some "average" Sketch Model each time. This would have been the case had the measure put into the ANOVA resulted from comparing the Sketch Models from different problem difficulty levels against each other. However, since the ANOVA operated on accuracy measures comparing each Sketch Model versus the true detection rate surface, the statistical results are strongly indicative of Bayesian behavior and are not indicative of a central tendency effect.

C. RECOMMENDATIONS

The following recommendations are made with respect to Sketch Model research in general and with respect to the Operational Decision Aids program specifically:

1. Repeat the air strike path experiment reported herein and correct for the two major weak points: subjects and trial problems. Train and test at least 20 subjects of appropriate rank and experience in the U.S. Navy. Redefine the problem set to ensure adequate difficulty for testing the impact of the Sketch Model on air strike utility performance.

2. Develop and implement Sketch Model smoothing by interpolation algorithms and measure the reduction in percent volume error.

3. Develop and implement the scalar subjective judgment anti-central tendency transformation and measure the reduction in percent volume error. Recommendations 2 and 3 can be carried out on the present data base without requiring another experiment.

4. Carry out a dedicated search for meaningful applications of the Sketch Model technique to naval task force commander decision problems. Identify the most promising of these and attempt to use the present data on percent volume error to predict decision performance in the candidate applications.

5. Develop the theory required to integrate the Sketch Model technique with the classical decision theoretic techniques which assume discrete states and discrete action/decision alternatives.

6. Investigate the use of three dimensional perspective (with and without hidden line capability) feedback in the development and use of Sketch Models. This may serve to reduce the central tendency error in the vertical dimension in the Sketch Model specification phase and encourage more intelligent use in the decision phase.

7. Develop and investigate methods that would allow the definition of time-varying Sketch Models (e.g., reflect impact of patroling sensors on time-varying detection rate contours).

8. The research reported here focused on only one of at least two ways to develop continuous subjective functions. An alternative to the Sketch Model approach is to provide the human with a limited set of functional forms and allow him to vary parameter values and combine different functional forms to define a continuous subjective function (CSF). Conduct a study to compare the Sketch Model technique to the parametric variation of a functional form.

9. Conduct a study to investigate the validity of the hypothesized advantages (Table 1) of the Sketch Model approach which were not addressed in the experiments reported in this document.

REFERENCES

1. Edwards, W. and Tversky, A., eds. Decision Making. Penguin, Baltimore, Maryland, 1967.

2. Irving, G.W., et al. ODA Pilot Study II: Selection of an Interactive Graphics Control Device for Continuous Subjective Functions Applications. Report 215-2. Integrated Sciences Corporation, Santa Monica, California. April, 1976.

3. Nemhauser, G.L. Introduction to Dynamic Programming, John Wiley and Sons, New York, 1967.

4. Payne, J.R. and Rowney, J.V. ONRODA Warfare Scenario. SRI/NWRC-RM-83, Stanford Research Institute, Menlo Park, California. June, 1975. AD# AO 16625.

5. Slovik, P., Fischhoff, B. and Lichtenstein, S. Behavioral Decision Theory. Technical Report DDI-7, Oregon Research Institute, Eugene, Oregon. September, 1976.

6. Winer, B.J. Statistical Principles in Experimental Design, New York, McGraw-Hill, 1971.

APPENDIX A

THE "WHERE AM I" ALGORITHM

APPENDIX A: THE "WHERE AM I" ALGORITHM

The "Where Am I" algorithm was developed to determine the detection rate at the end points of a strike path leg with respect to Sketch-Modeled contours. Given that we have a set of closed contour curves, each representing points at a certain altitude level, and an arbitrary point in the 2-D plane, we are to determine what altitude level the point is in.

The method used is to shoot out a ray in one direction from the point of interest and record the contour crossings encountered by the ray. The contour crossings are arranged in ascending order according to their distances from the point of interest. This list is to be checked for double crossings for each altitude from the closest crossing to the furthest crossing, and all double crossings are deleted from the list. Upon completion, the altitude at the closest single crossing is the altitude level the point is on. The rationale is that an odd number of crossings on a closed curve will occur if any only if the point is enclosed by this curve.

Two examples are illustrated in Figure 50. From point A we shoot out a ray in the Y direction, and the contour crossings, in ascending order according to their distances from A, are 4, 5, 5, 4, 3, 2, 1.

Deleting double crossings from the previous list, we have 3, 2, 1.

The closest crossing in the revised list tells us that point A is at altitude level 3.

As a second example, consider point B. Shooting out a ray from B in the +Y direction yields the following list of contour crossings: 3, 3, 2, 1.

After deleting the double crossing from our list, we have 2, 1.

Hence, we know point B is an altitude level 2.

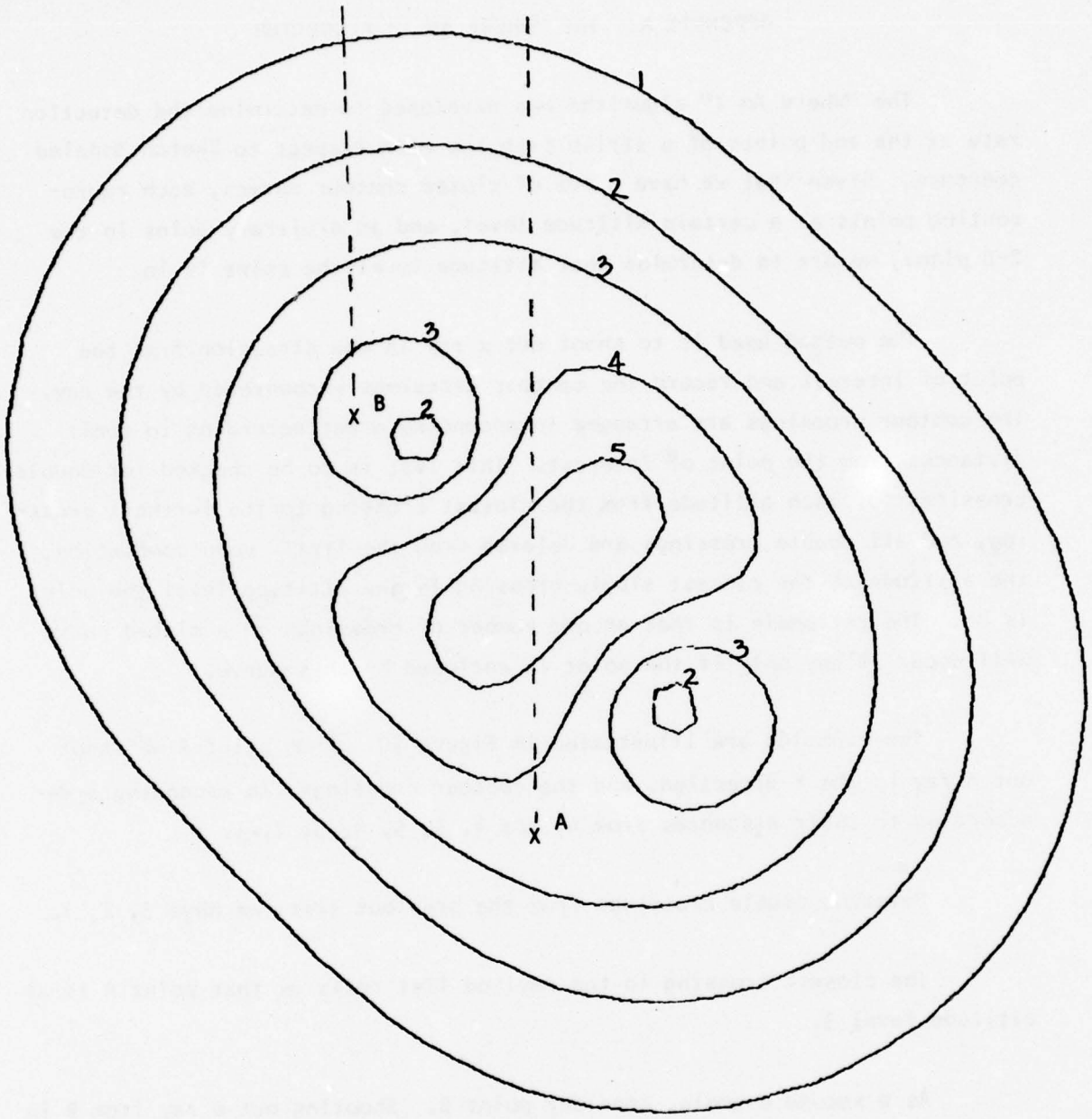


Figure 50. Boundary Crossing Examples.

Notice that the rays can be in any direction. The +Y direction is chosen in our examples, but we could just as well have chosen the -Y direction, or any other direction.

To extend the contour crossing method to situations where the contour curves are not closed, we note that the only restriction we have to impose is that the domain boundary line in the path of the ray we shoot out does not intersect any contour. Hence the situation depicted in Figure 51 can be handled by our "Where Am I" contour crossing algorithm. As long as the rays are in the +Y direction, the algorithm gives correct results for the example in Figure 51.

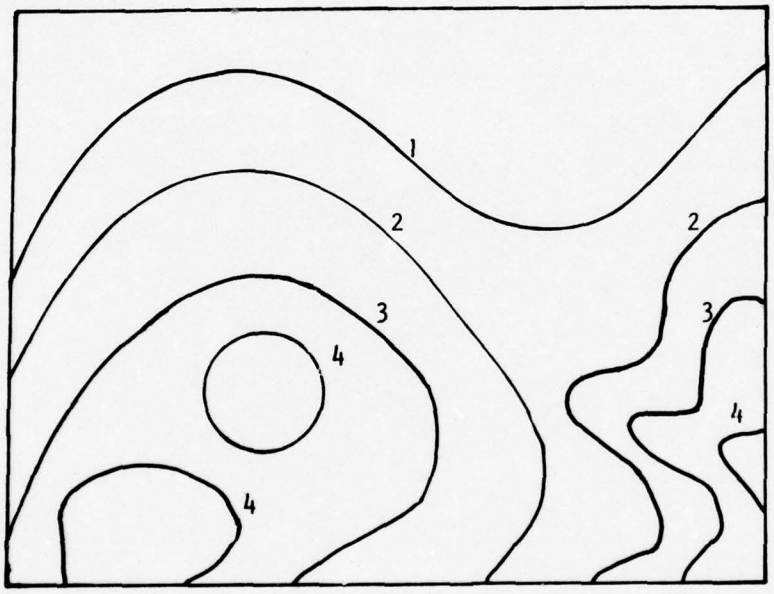


Figure 51. An Example of a Domain Boundary Line Clear of Intersections with Contours.

Inherent in our algorithm is the requirement to detect contour crossings for the ray we shoot out from the point of interest. Since each contour is defined by a string of adjacent points on the curve, we can examine two adjacent points on the contour at a time. If there is a crossing, it must occur between one of these pairs of points. By successively checking each

pair of adjacent points on a contour, we can thus detect all crossings. In actuality it is not always possible to determine a crossing from a pair of adjacent points alone. Information about preceding or succeeding pairs of points on the contour may be needed. A detailed flowchart for the algorithm is given in Figure 52. There we have arbitrarily chosen the +Y direction for shooting a ray out from the point of interest.

In the flowchart all the possible orientations (shown in Figure 53) for a pair of adjacent contour points and the point of interest are accounted for.

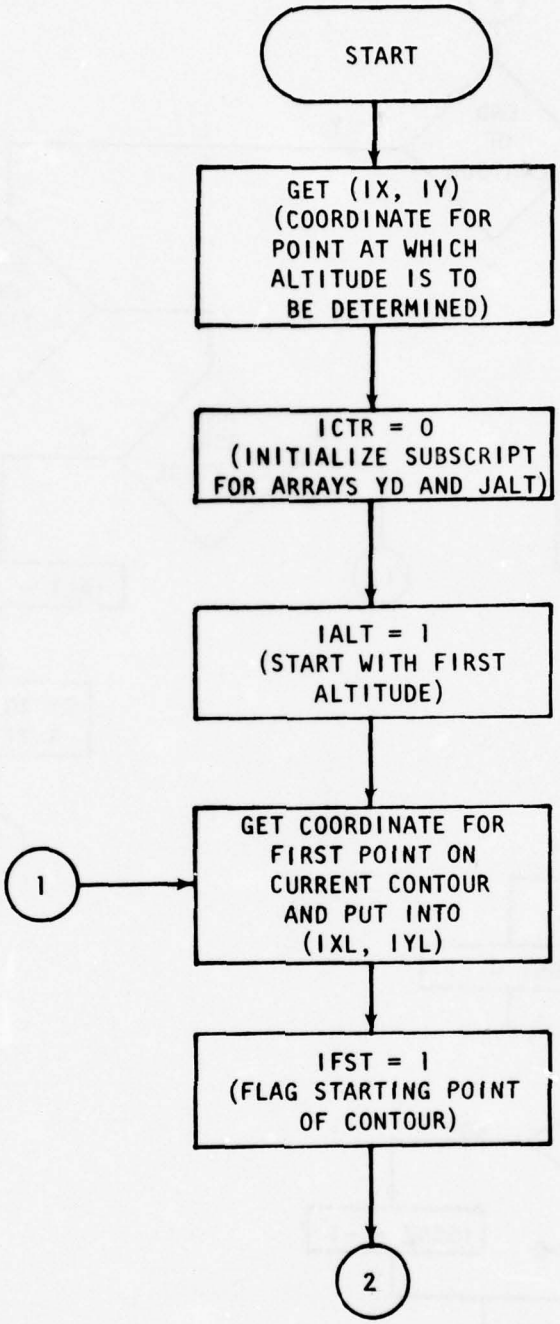


Figure 52. Flowchart for Contour Crossing Algorithm. Case Numbers Correspond to those in Figure 53.

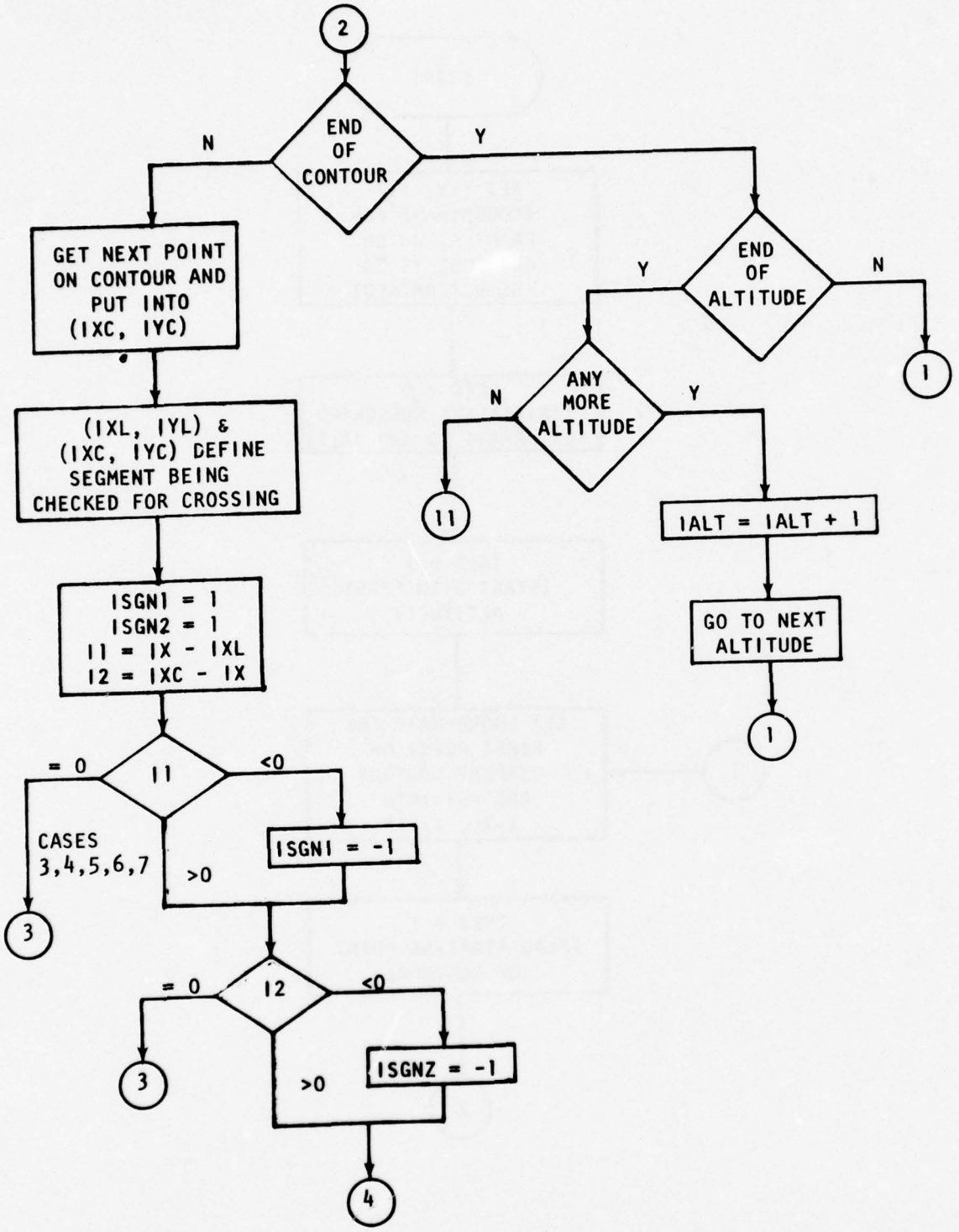


Figure 52. (Continued).

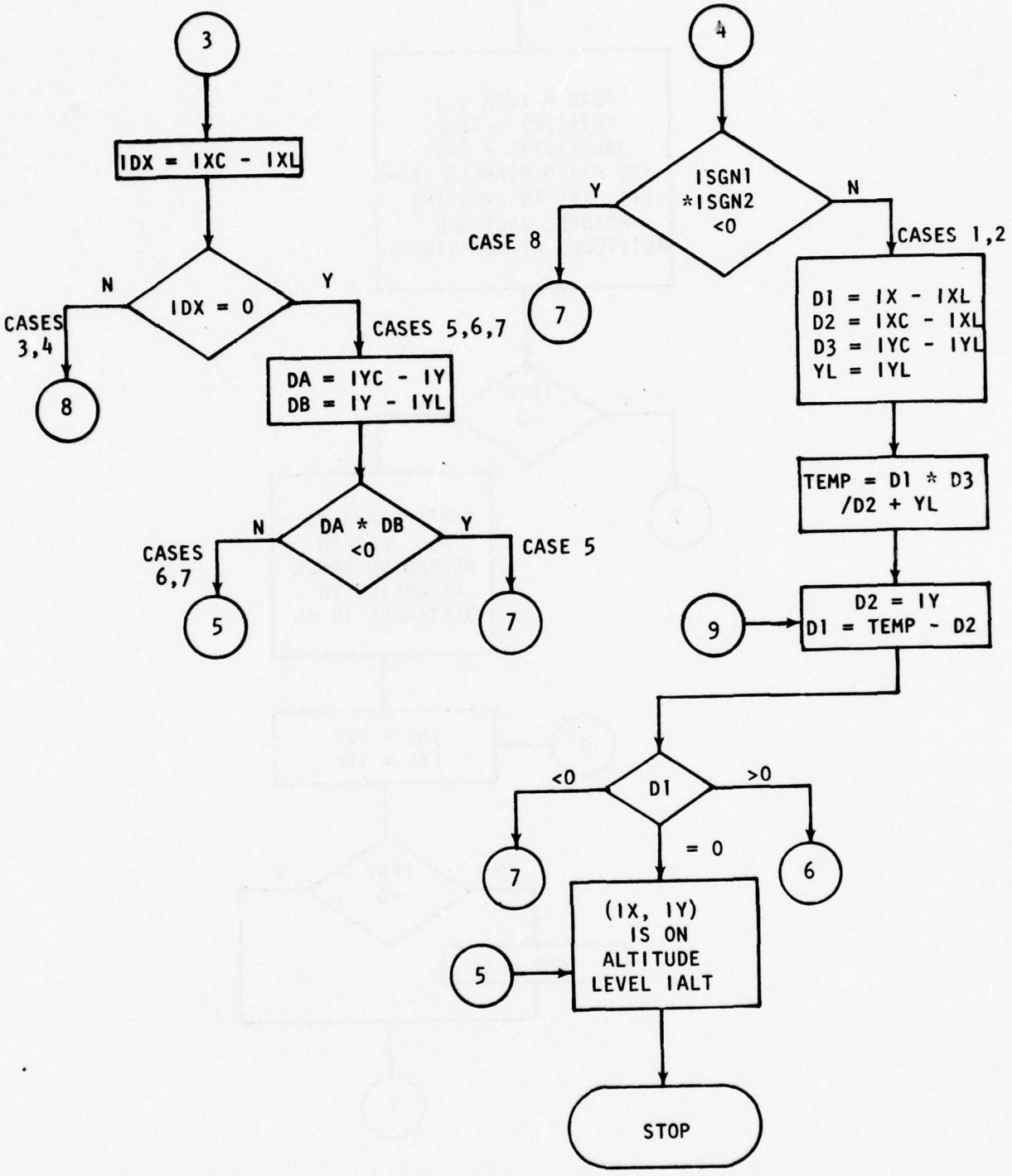


Figure 52. (Continued).

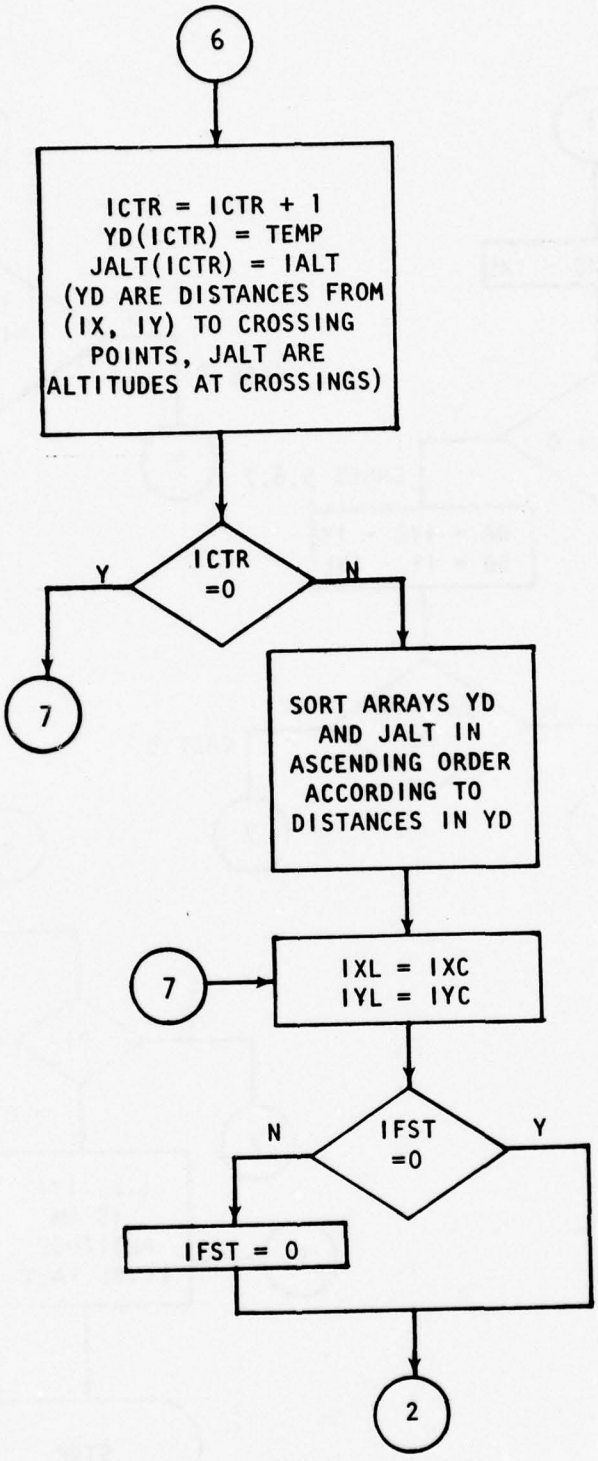


Figure 52. (Continued).

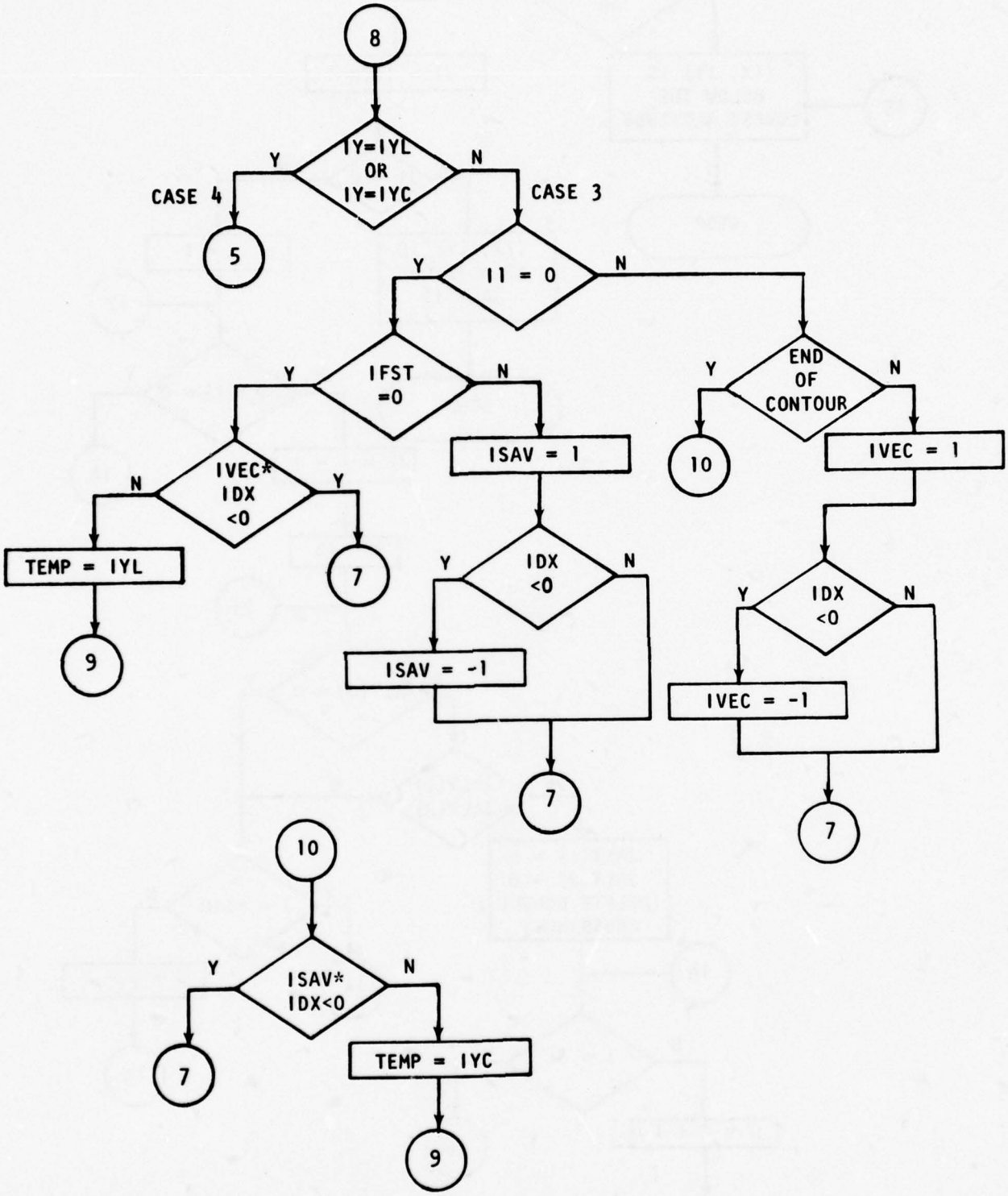


Figure 52. (Continued).

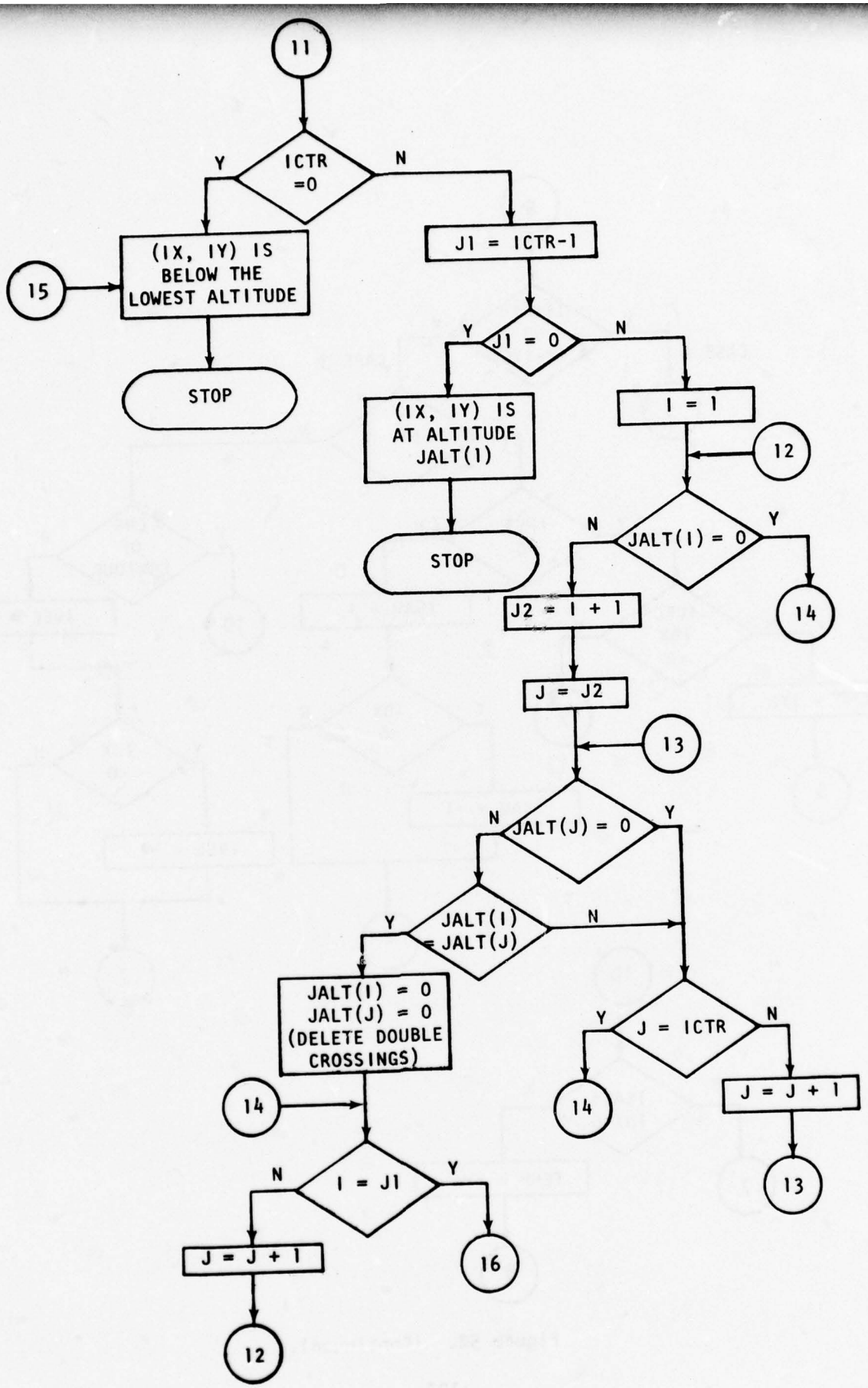


Figure 52. (Continued).

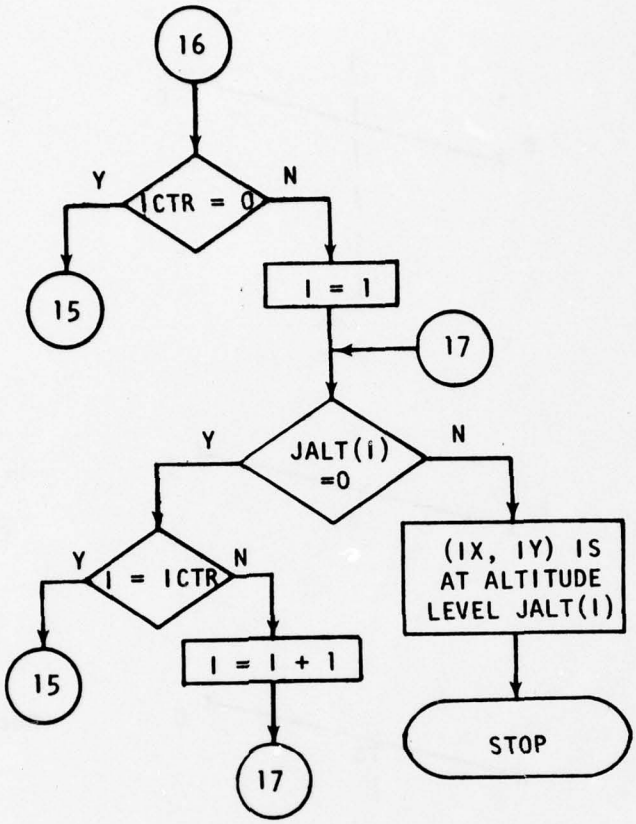


Figure 52. (Continued).

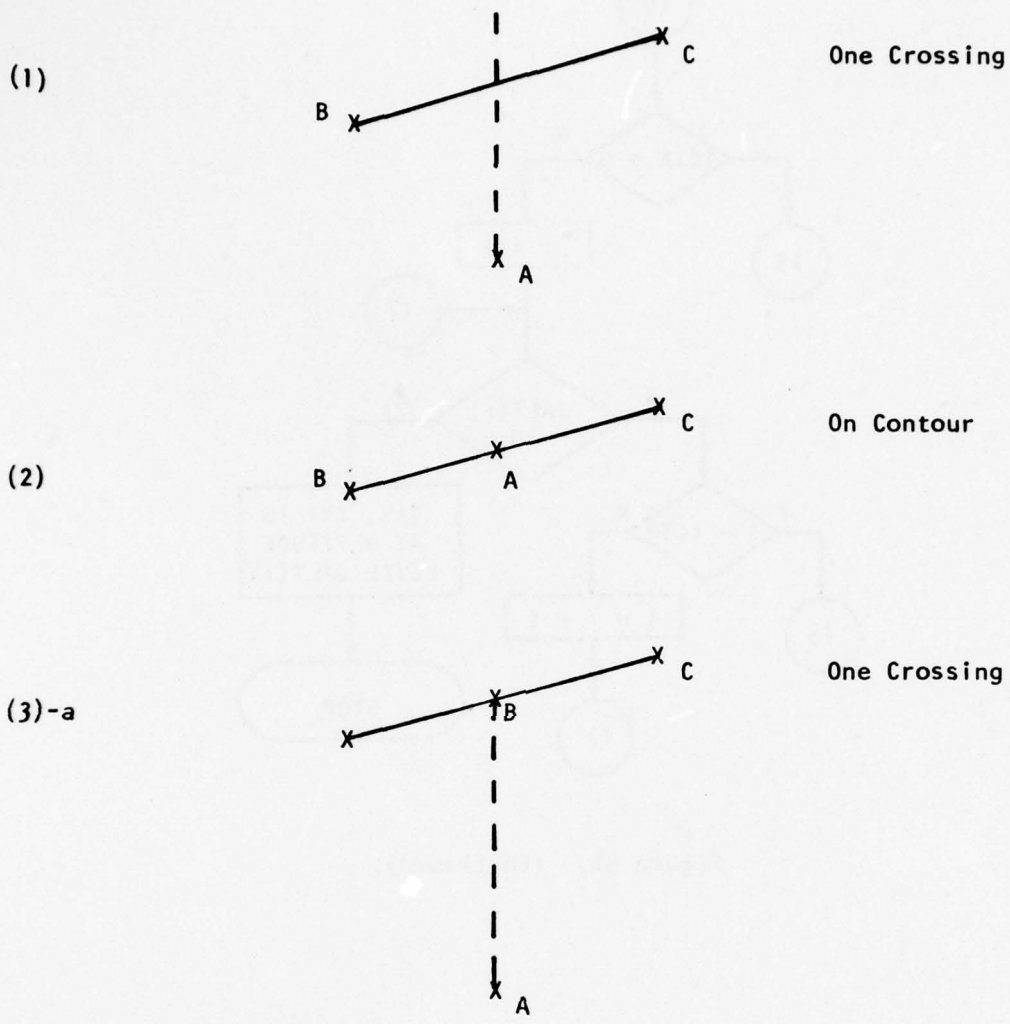
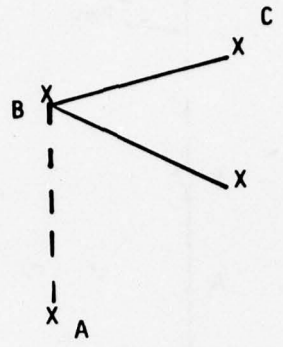


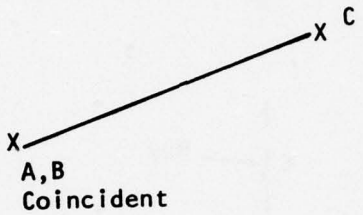
Figure 53. Enumeration of Possible Orientations for Points A, B and C. A is the point at which we desire to determine the altitude. B and C are adjacent points on the contour, which is defined by straight line segments connected between adjacent points. The ray shot out from A is indicated by a dashed line.

(3)-b



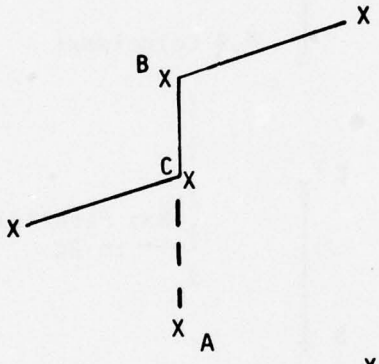
No Crossing

(4)



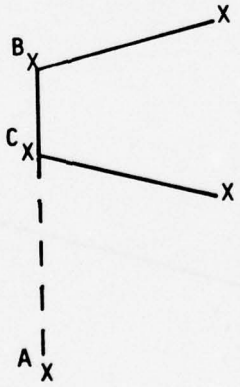
On Contour

(5)-a



One Crossing

(5)-b



No Crossing

Figure 53. (Continued).

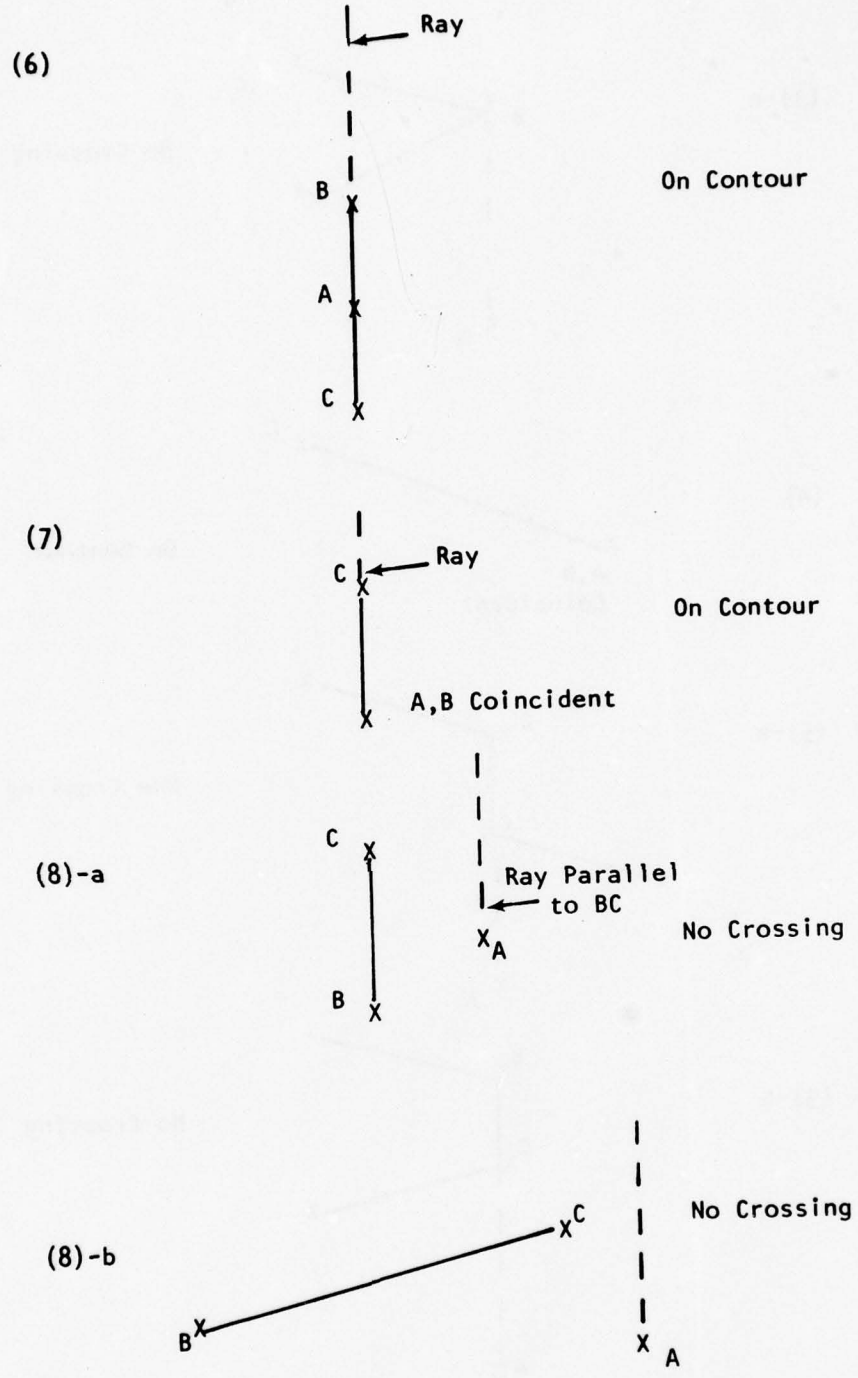


Figure 53. (Continued).

APPENDIX B

TRUE CONTOUR DRAWING ALGORITHM

APPENDIX B: TRUE CONTOUR DRAWING ALGORITHM

For the experiment, it was necessary to come up with a set of "true" composite detection rate contours for each configuration of sensor locations and sensor sizes used. Each set of true contours corresponds to a "Sketch Model" the subjects were required to draw, and the true contours were displayed as feedback to the subjects at the end of each trial under System Concept B. The development of the contour following algorithm which traces out the contours required is described below.

A three-dimensional function $f(x,y)$ defined in a region $x_o \leq x \leq x_t$, $y_o \leq y \leq y_t$ can be represented in two-dimensional space as a series of iso-altitude contours. In other words, the curve $f(x,y) = h$ can be plotted for every altitude h we have chosen. Each contour thus represents points at the same specified altitude.

The contour following algorithm requires two major steps. In the first step, points at each desired altitude level are sampled and stored in a list. Then each list of points is examined and the points connected so that the resulting curves describe the true contours reasonably well. These two steps are further explained below.

To simplify the exposition, let us for the moment concern ourselves with one altitude level only. The method is easily extended to more than one altitude.

Suppose we want to find the contours for an altitude h. First we search in the x-y domain for points at this altitude. We try to find a sufficient number of points (x,y) such that $f(x,y) = h$. This is done in two passes.

In the first pass, a flowchart of which is given in Figure 54, we lay down grid lines δ units apart in the Y direction. The value δ is selected such that the grid lines are close enough for the resulting contour

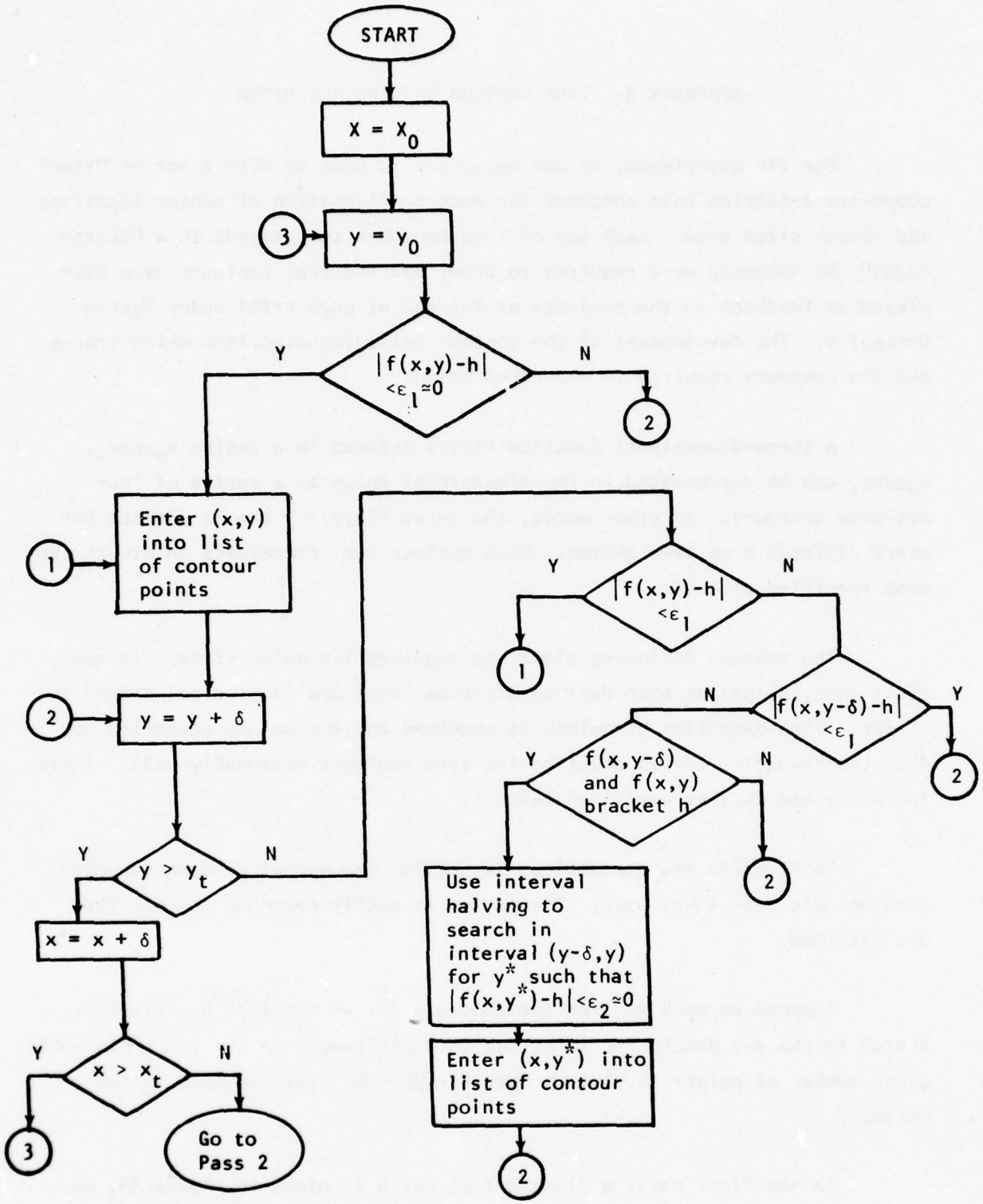


Figure 54. Flowchart for First Step of Contour Following Algorithms, Pass 1.

points not to be too sparse. At the same time the grid lines are not so close that excessive time is spent in the computations. Each grid line is then divided into intervals of width δ (see Figure 55). This leads to another consideration for the value of δ in that the finer the grid resolution the easier it is to detect sharp ridges and peaks. The functional values at the end points of each interval are checked to see if the desired altitude h is bracketed. In other words, we check to see if h is between the values of the function at the two end points. If bracketing occurs we know we can find a point within the interval where the value of f is sufficiently close to h. We do this by successively halving the interval until we are close enough to the desired point. Each point on the contour found is put into a list for later use.

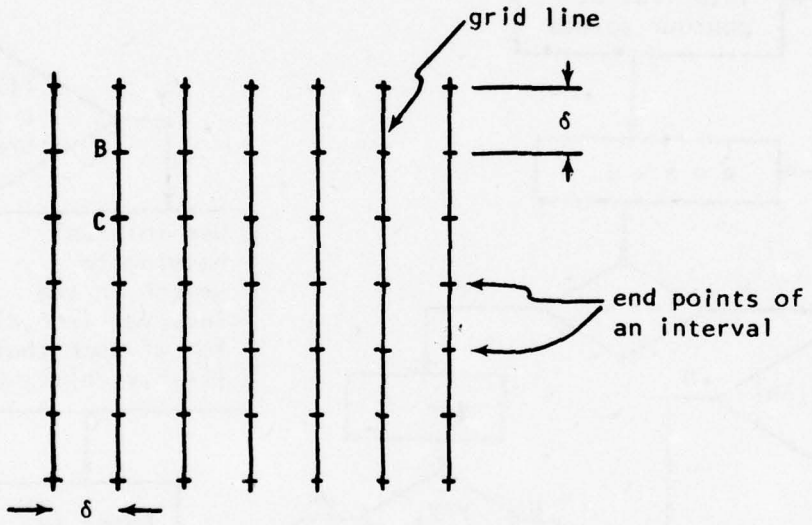


Figure 55. Example of Grid Line Division. (Each grid line is divided into intervals, such as BC.)

At the completion of the first pass, we move to the second pass; the corresponding flowchart is shown in Figure 56. Here we lay down a second grid of the same grid size δ, but this time the grid lines run in the X direction. Each grid line is divided into intervals, each of length δ. This second grid is necessary because the first grid cannot "see" very well those contours that are parallel to the grid lines, as shown in Figure 57.

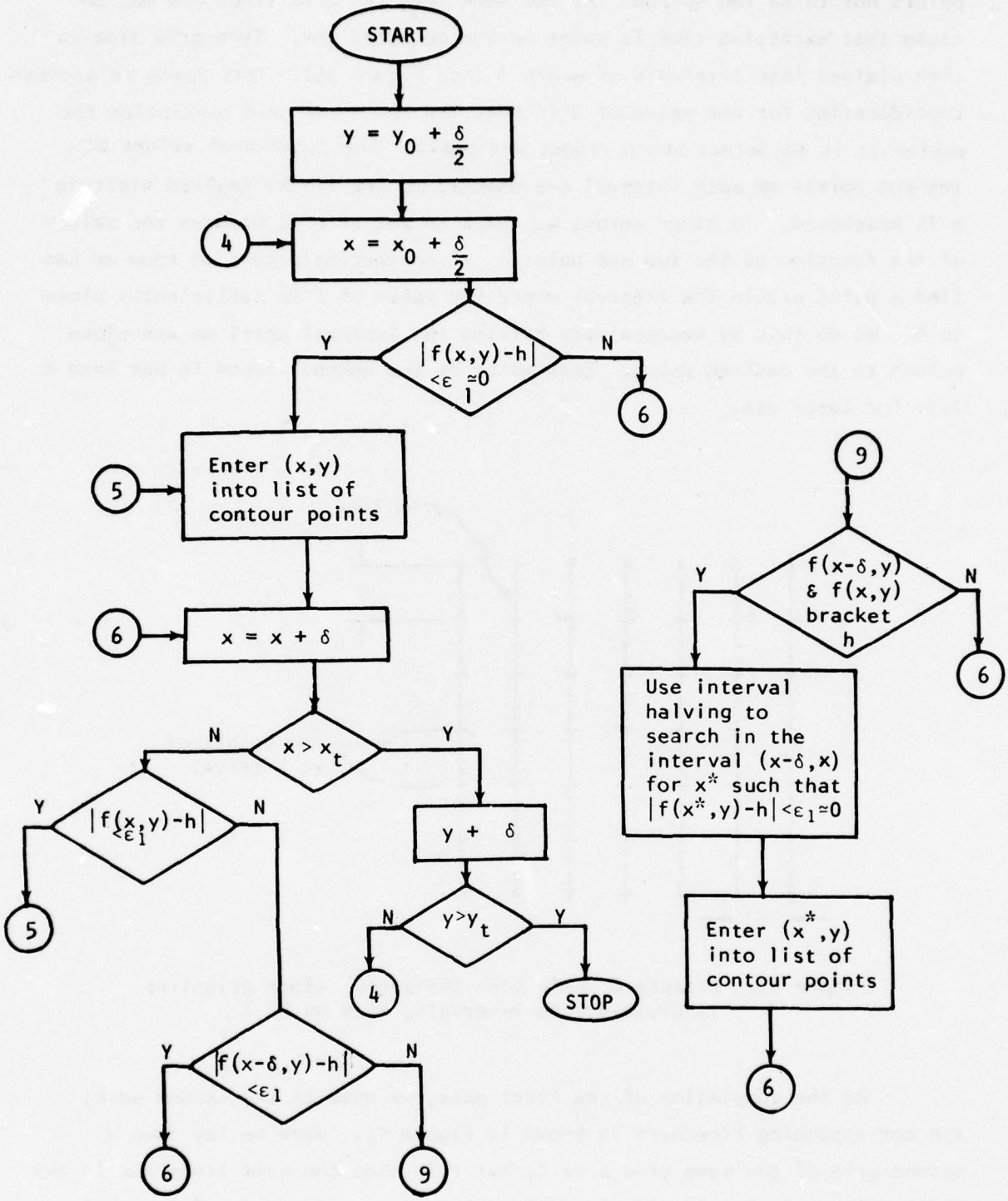


Figure 56. Flowchart for First Step of Contour Following Algorithm, Pass 2.

Since the second grid is perpendicular to the first, it can be used to remedy problems such as that depicted in Figure 57. The second grid is arbitrarily displaced from the first so that grid points from the two will not overlap, but this is not necessary.

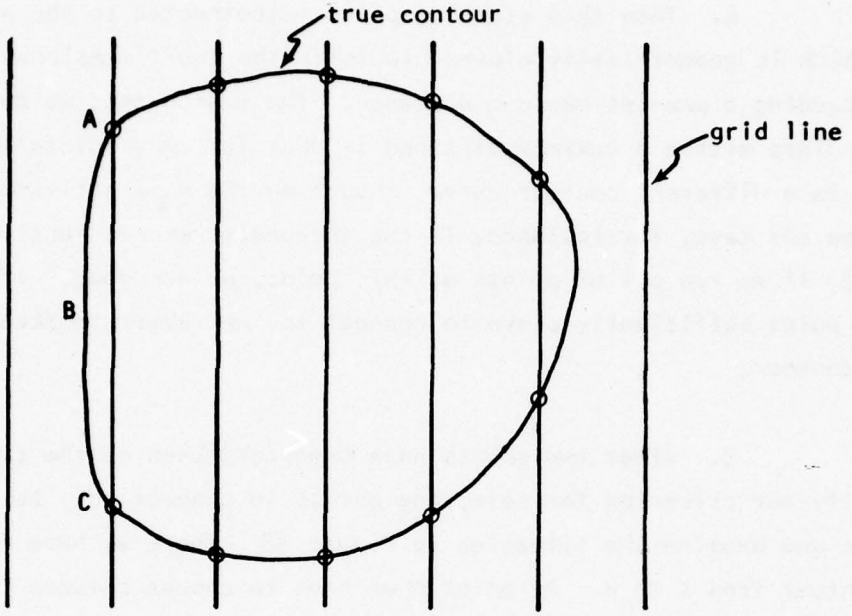


Figure 57. Error When Using a Single Grid. (With grid lines in the shown direction, the ABC portion of true contour is not detected by the grid lines. The dots are contour points picked up by grid.)

With the second grid, each interval on a grid line is checked for bracketing of the desired altitude h. Each occurrence signifies a contour point within the interval under consideration. Searching by interval halving yields a point on the contour, and this point is added to our list of contour points.

At the completion of the second pass we have a list of points at the altitude h. Obviously an empty list means that no contour at that altitude could be found. The next step is to reconstruct contours with this non-empty list of points.

We generate the contours by connecting the points in a meaningful manner. A simplified description of the algorithm used is given below.

1. To begin, a point is obtained from the previously generated list as a starting point for a new contour.

2. Then this starting point is connected to the point in the list which is geometrically closest to it in the two-dimensional plane, but not exceeding a pre-set maximum distance. The reason that we consider only those points within a maximum distance is that far away points most likely belong to a different contour curve, though on the same altitude. (This would be the case, for instance, if the three-dimensional function is bi-modal.) If we run out of points at this point, we are done. If we fail to find a point sufficiently close to connect to, we revert to Step 1 to start a new contour.

3. After two points have been connected on the current contour, we modify our criterion for selecting points to connect to. Let us pause for a while and examine the situation in Figure 58. There we have just connected the contour from A to B. At point B we have to choose between C and D as the next point to connect to. Point D is closer than point C, but going from B to D involves a greater directional change, as measured by $\Delta\Theta_2$ in the figure. Assuming our contour to be a smooth curve with minimal zig-zagging, we may want to connect to point C instead. In order to accomplish this aim, the criterion function for selecting the next point is modified to be

$$\Delta S + \beta \cdot \Delta\Theta_1$$

where

ΔS is distance to candidate point (e.g., the length BC)

$\Delta\Theta$ is the directional change (e.g., $\Delta\Theta_1$)

β is the weighting factor to be selected by trial and error to yield acceptable results for types of curves under consideration

With this modified "distance" criterion, we always select a point with the smallest criterion function. As before, only those points which lie within a sufficiently small distance will be considered. Before

connecting to the point selected, it is compared with the starting point (obtained in Step 1), if the starting point is close enough. Between the two, the one with the smaller criterion function is the next point to connect to. This provision allows us to close the contour curve when we reapproach the starting point.

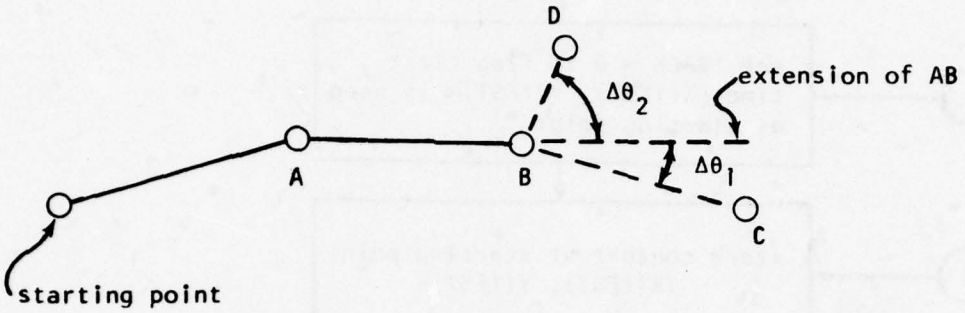


Figure 58. Selection of Candidate Points for Next Point on Contour.

4. Whenever the contour is closed by reconnecting to the starting point, we revert to Step 1 to start a new contour.

5. If in Step 3 we could not find a point close enough to connect to, we have an open curve. (An open curve can occur, for instance, when part of the composite detection field lies outside the frame representing the situation geography boundaries. Examples can be readily seen in Appendix C.) In this case we return to the starting point for the current contour and search for another point close to the starting point to connect to, by reverting to Step 2. For each open contour this procedure of reverting to the starting point will be carried out only once. When nothing can be done to close the contour, we assume it is an open curve, and we go back to Step 1.

6. Note that in any step a point already connected to will no longer be considered. The algorithm stops when we are out of points.

Figure 59 provides a more detailed representation of the algorithm.

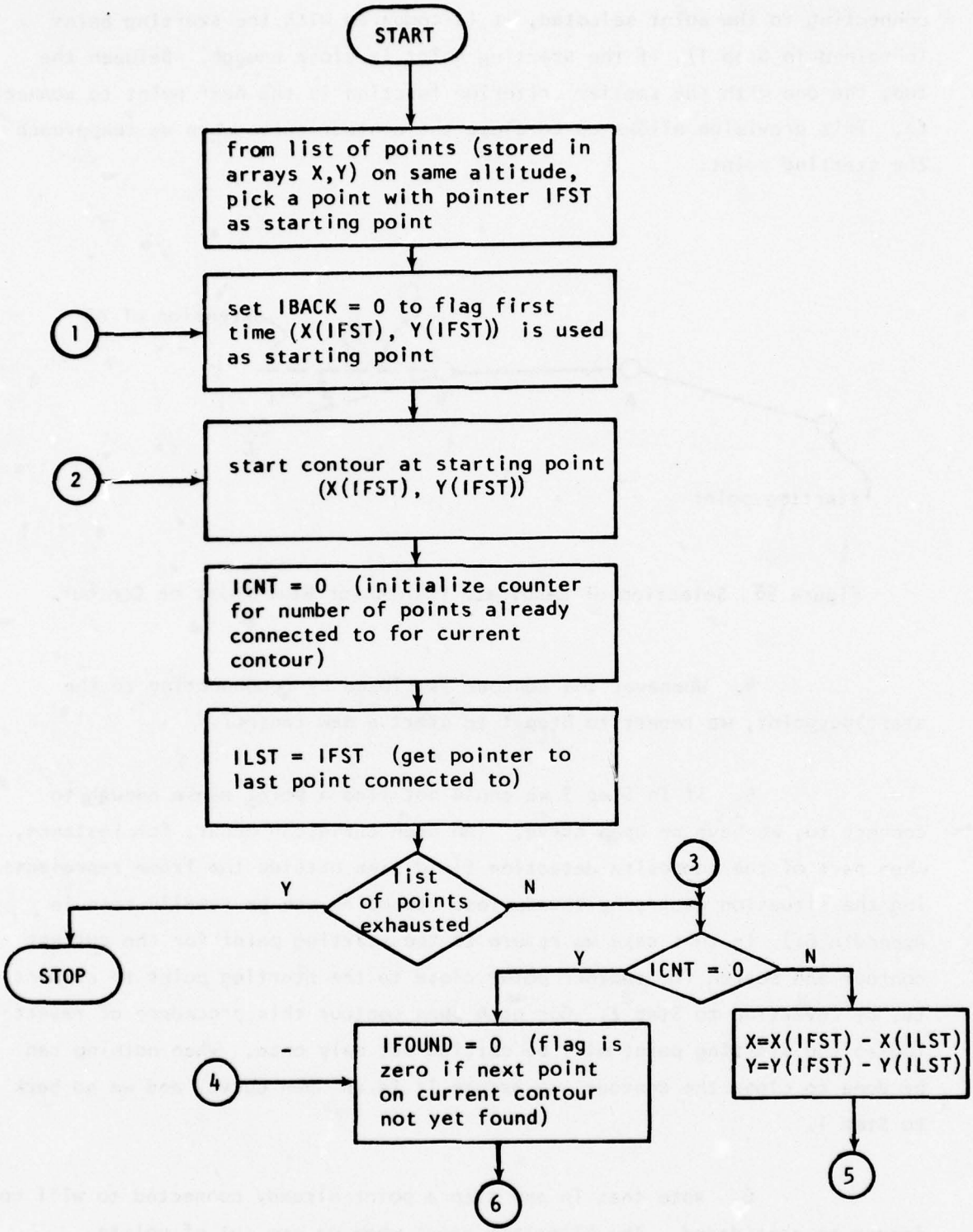


Figure 59. Flowchart for Contour Following Algorithm, Second Step.

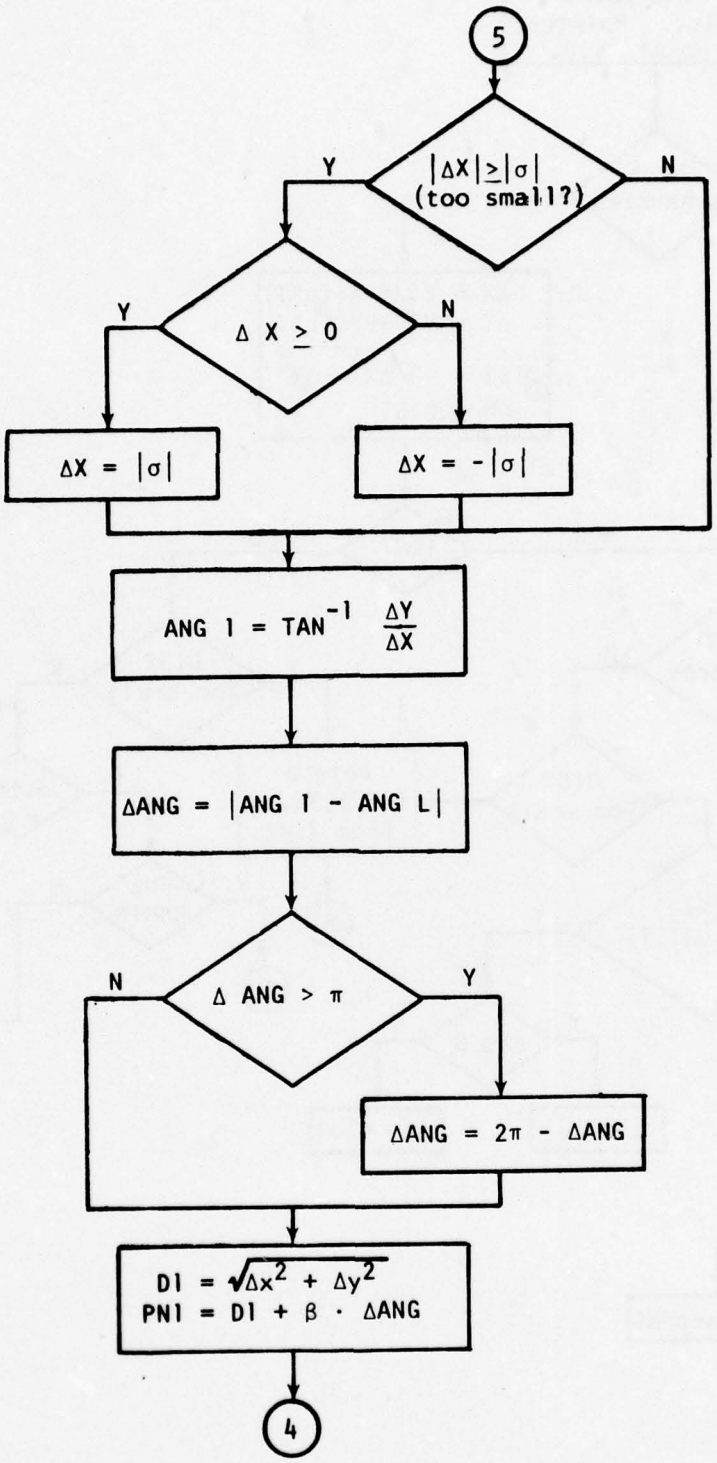


Figure 59. (Continued).

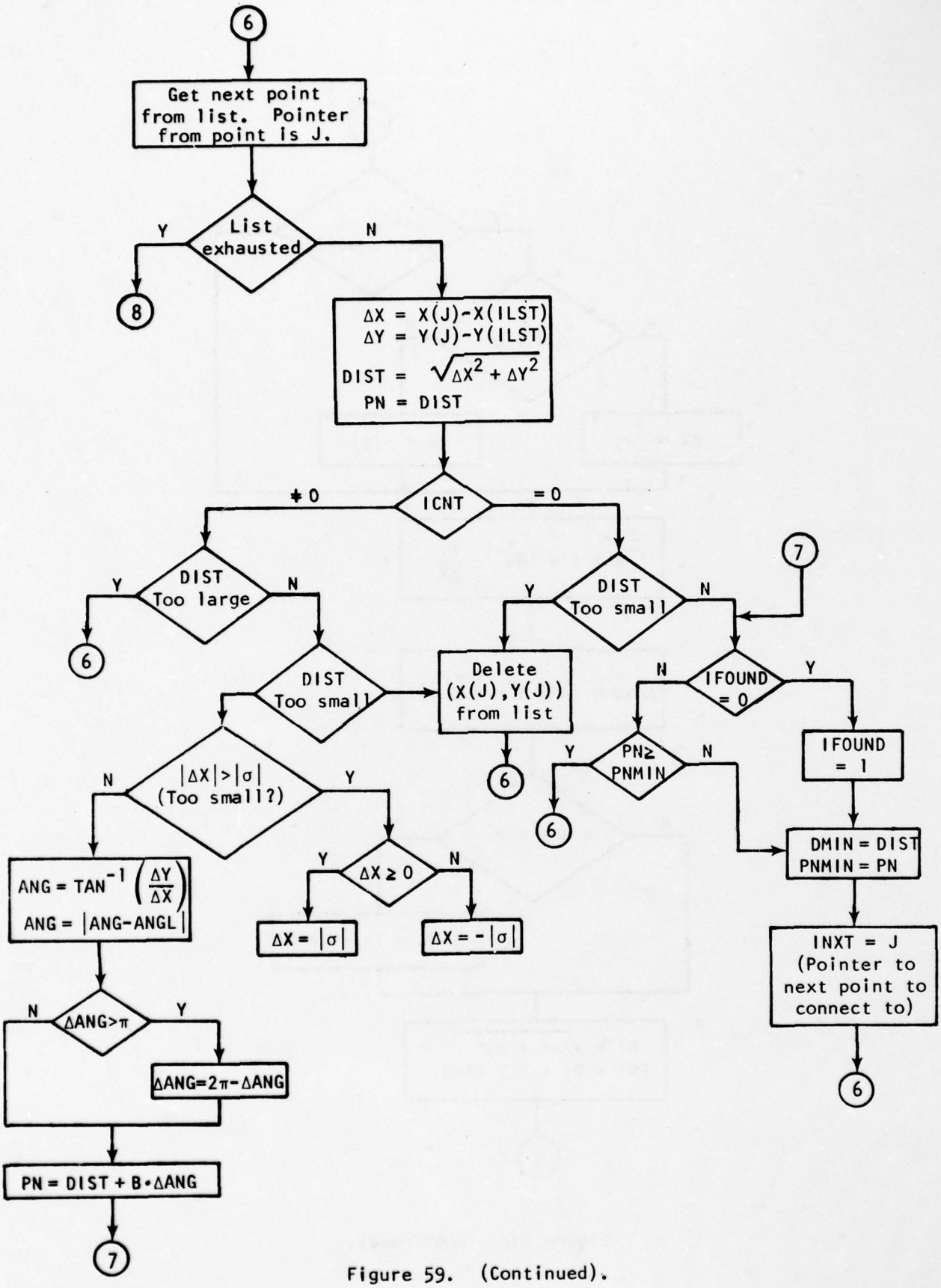


Figure 59. (Continued).

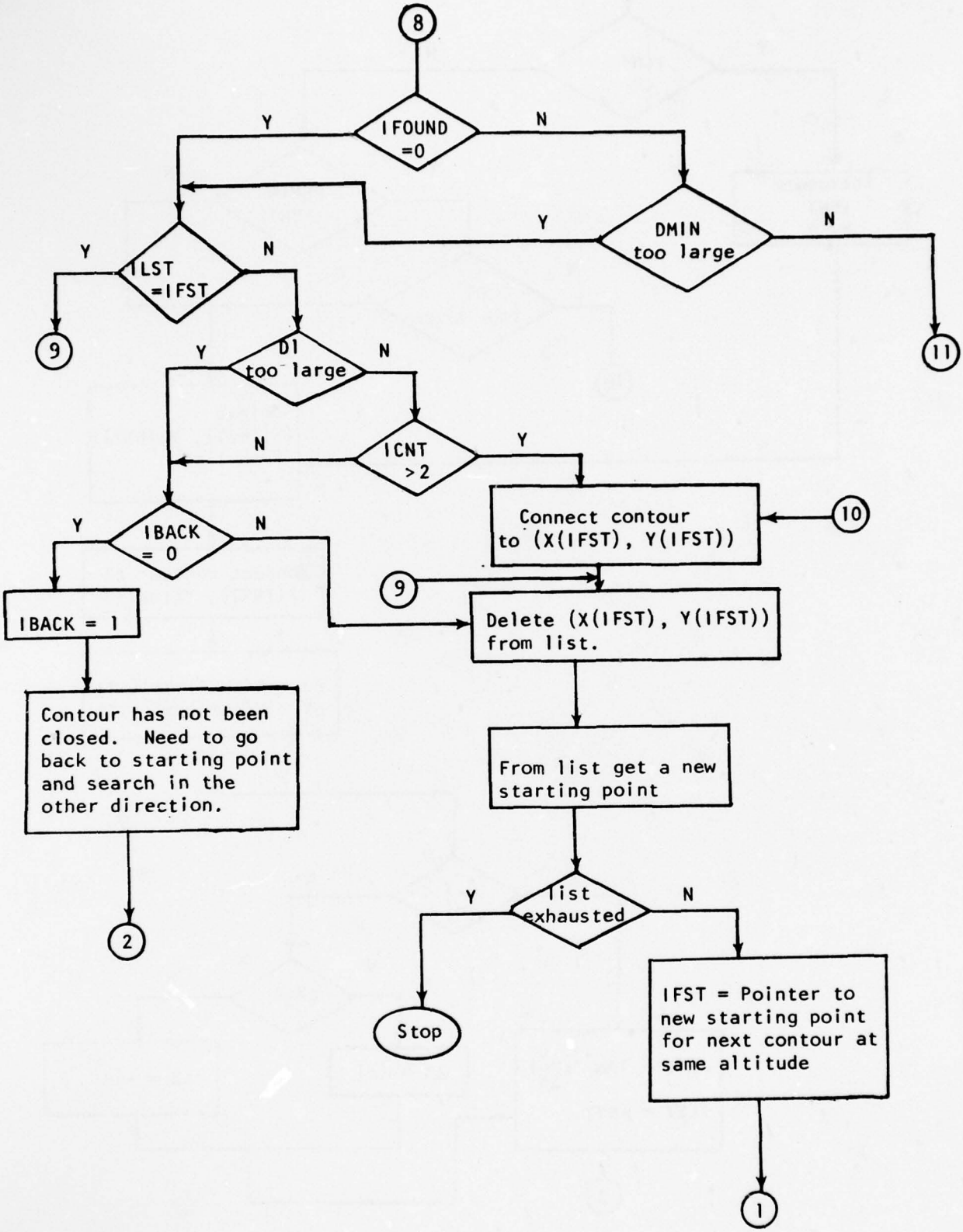


Figure 59. (Continued).

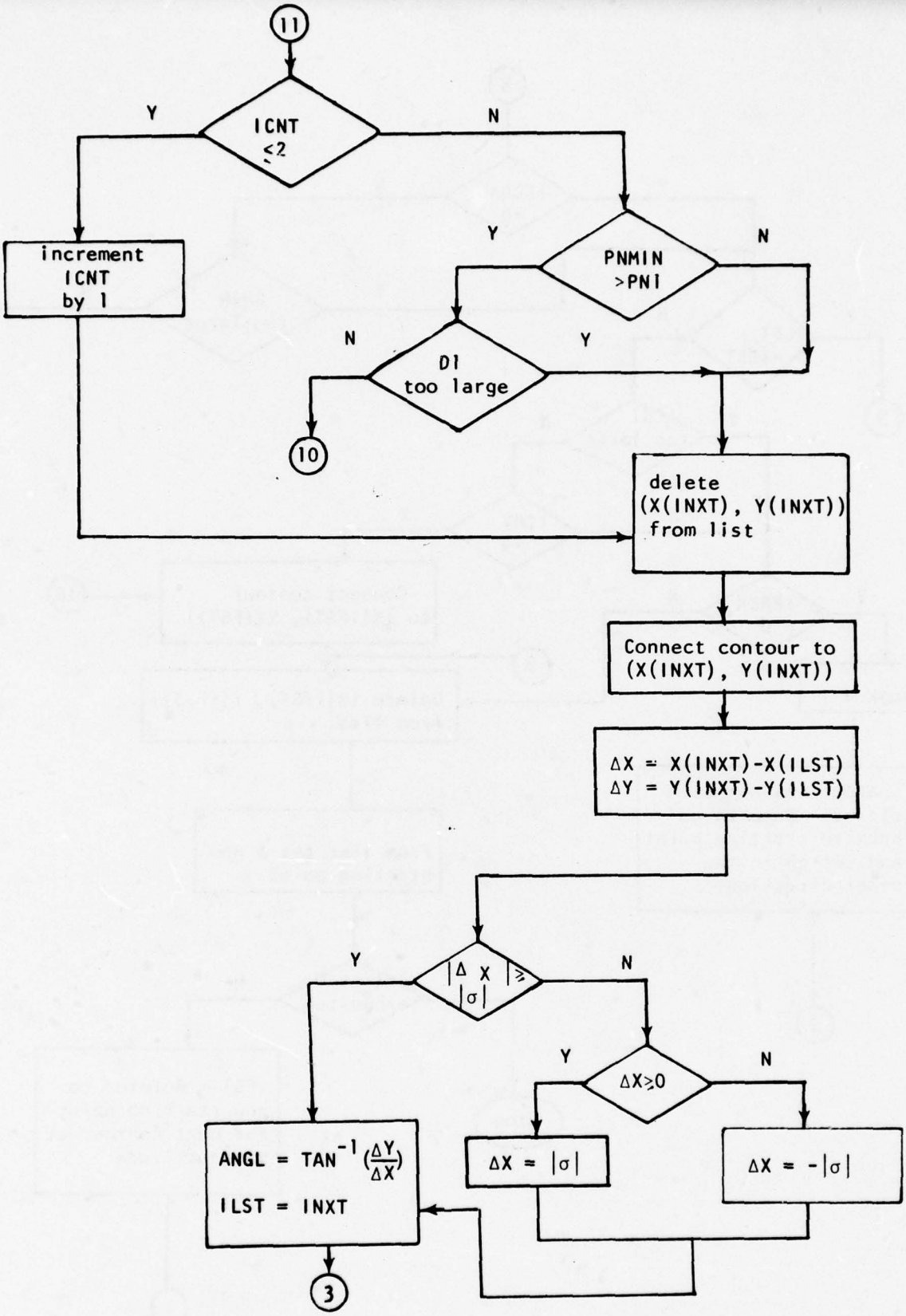
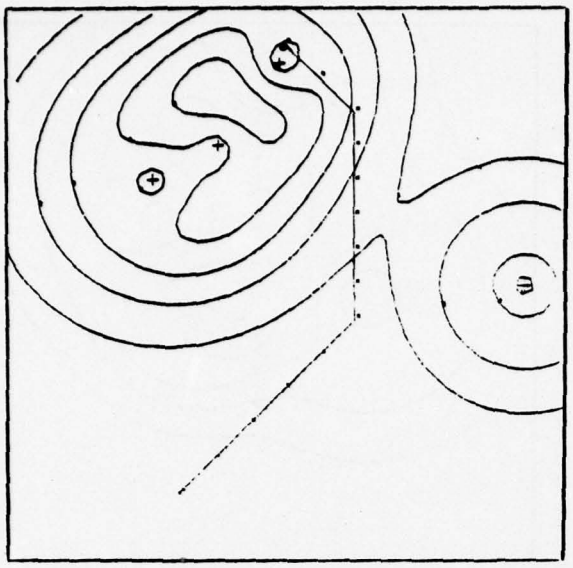


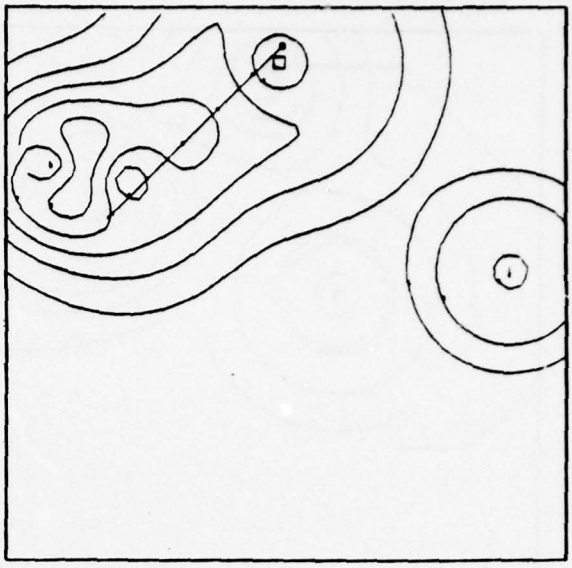
Figure 59. (Continued).

APPENDIX C

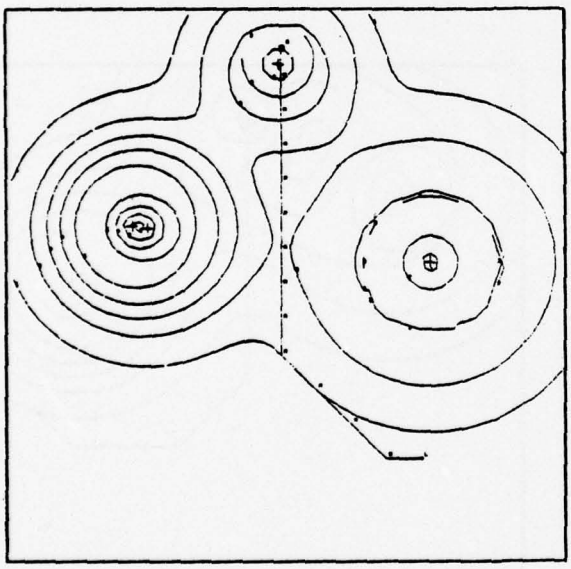
TRUE COMPOSITE DETECTION RATE CONTOURS



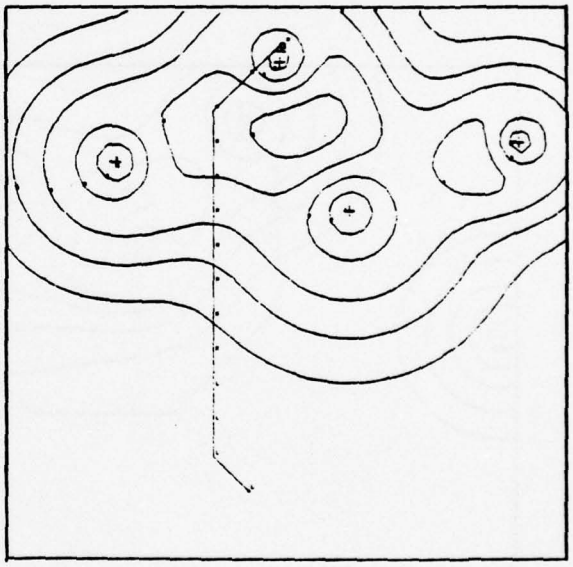
Problem 1



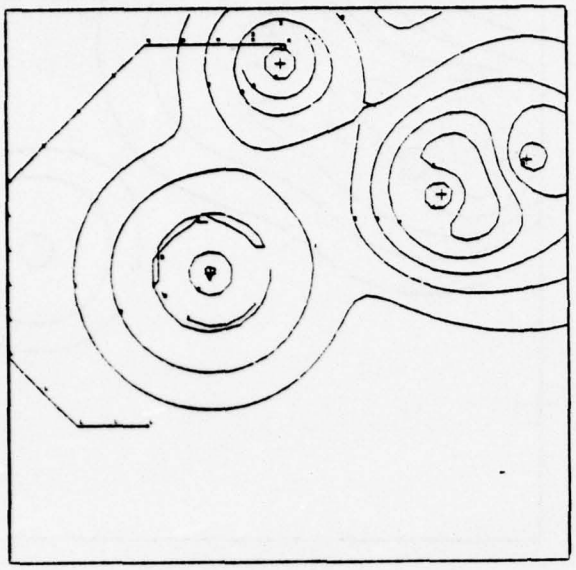
Problem 2



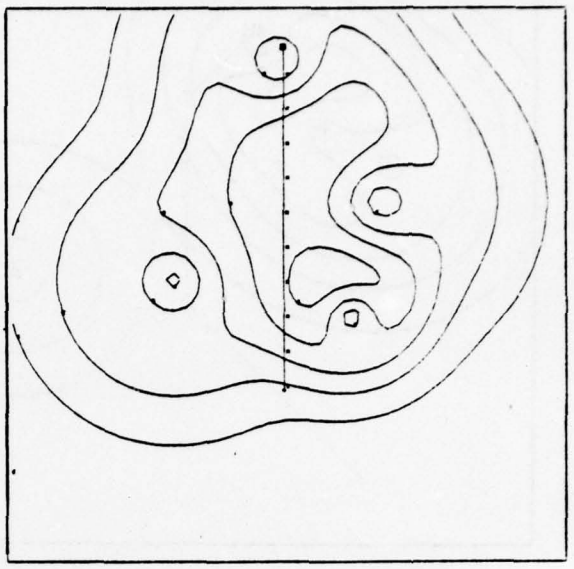
Problem 3



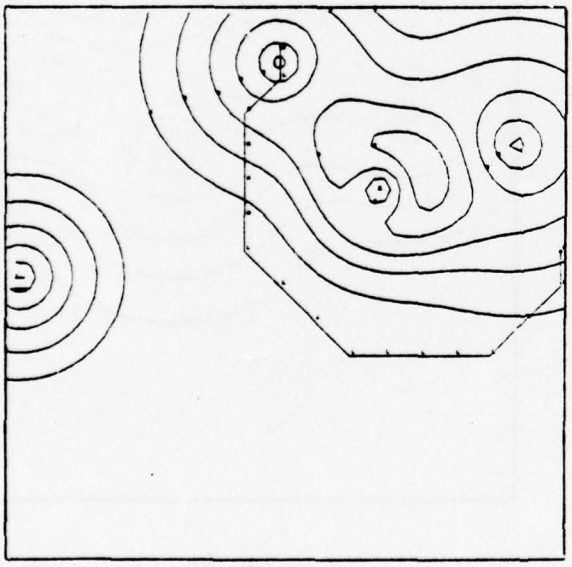
Problem 4



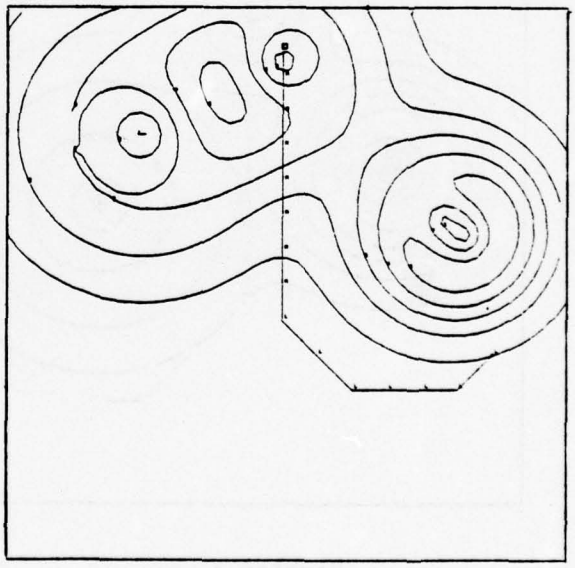
Problem 5



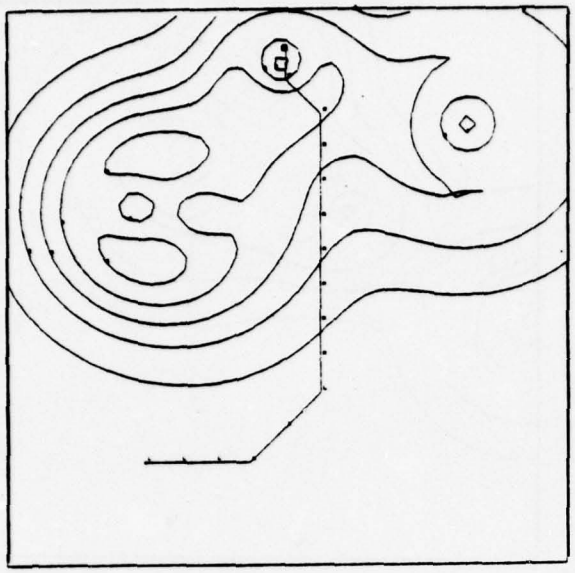
Problem 6



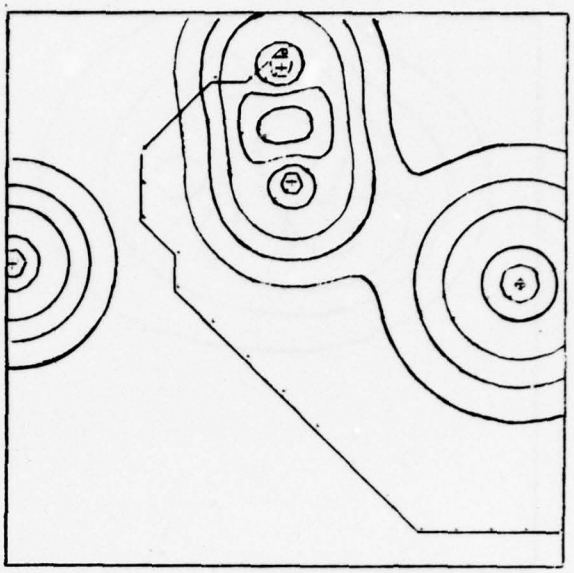
Problem 7



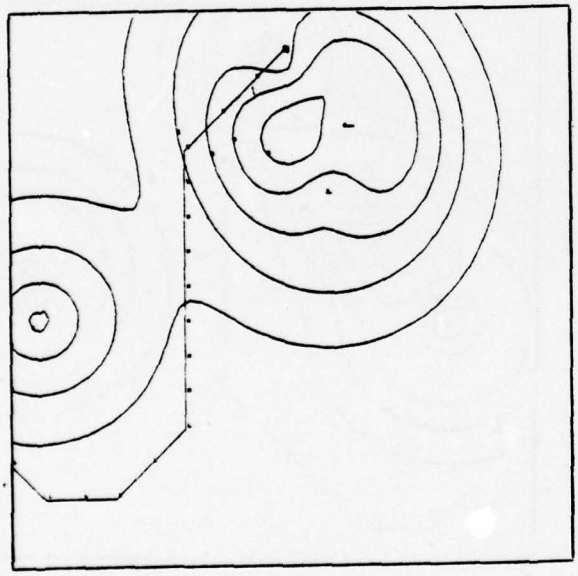
Problem 8



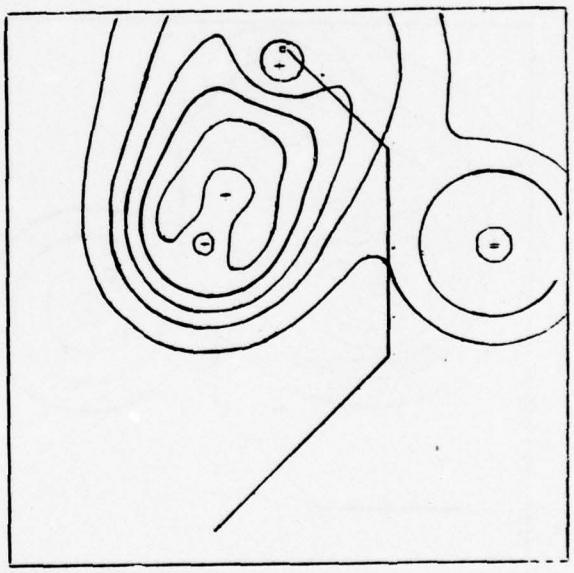
Problem 9



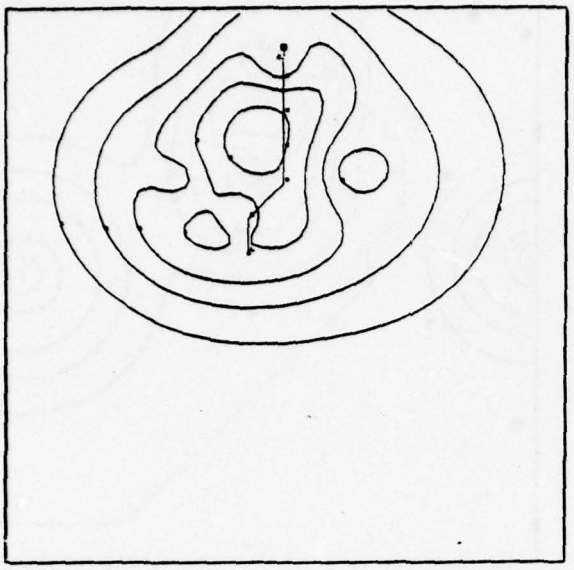
Problem 10



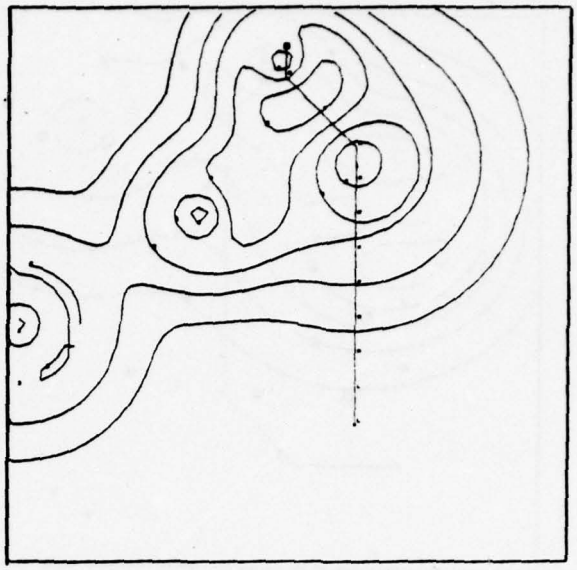
Problem 11



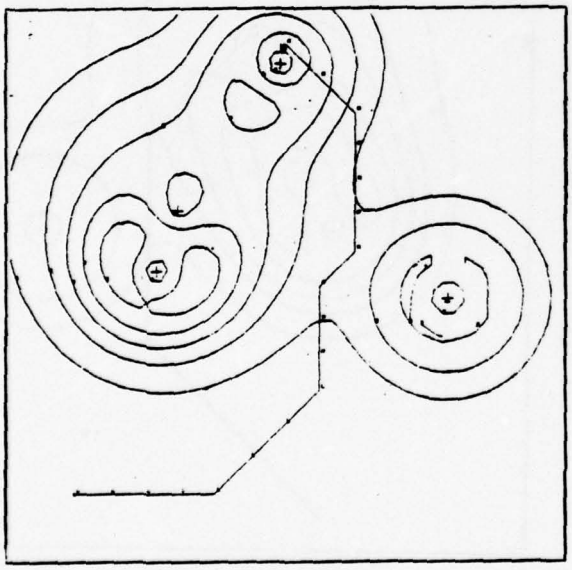
Problem 12



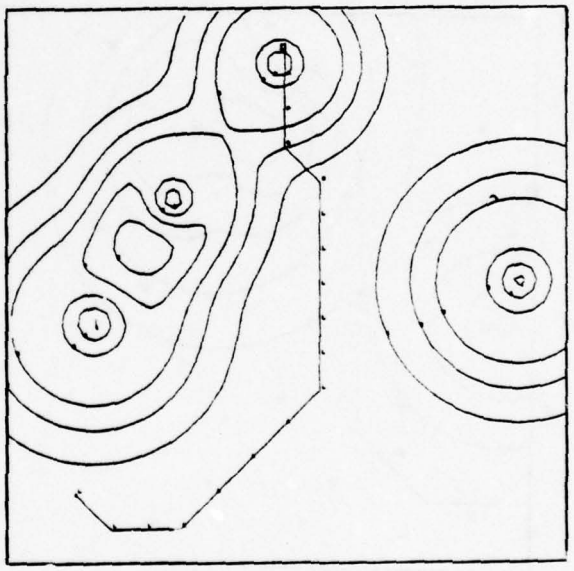
Problem 13



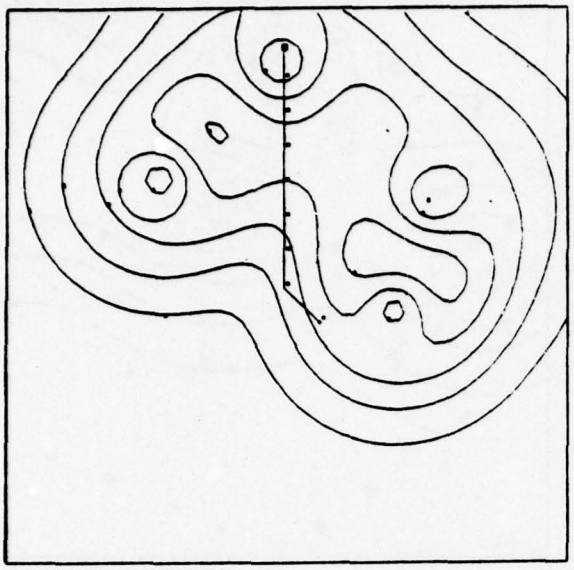
Problem 14



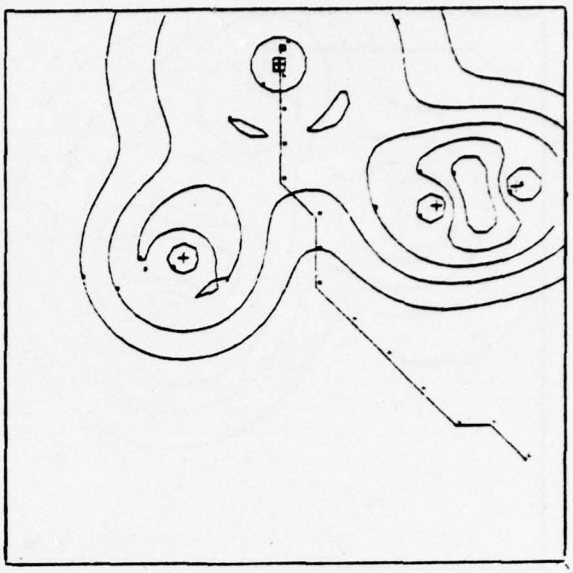
Problem 15



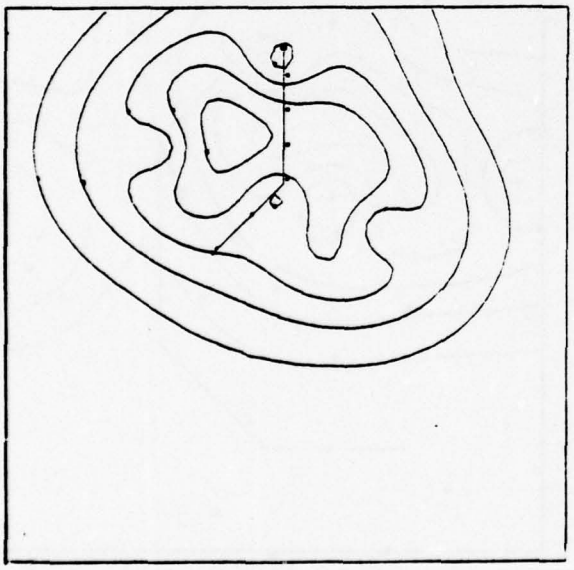
Problem 16



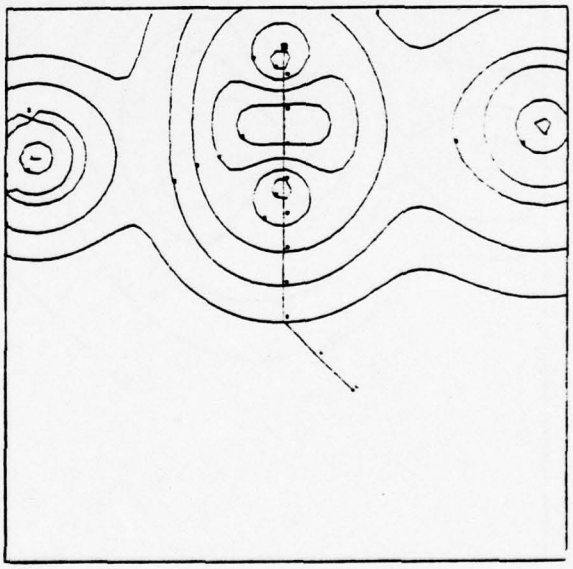
Problem 17



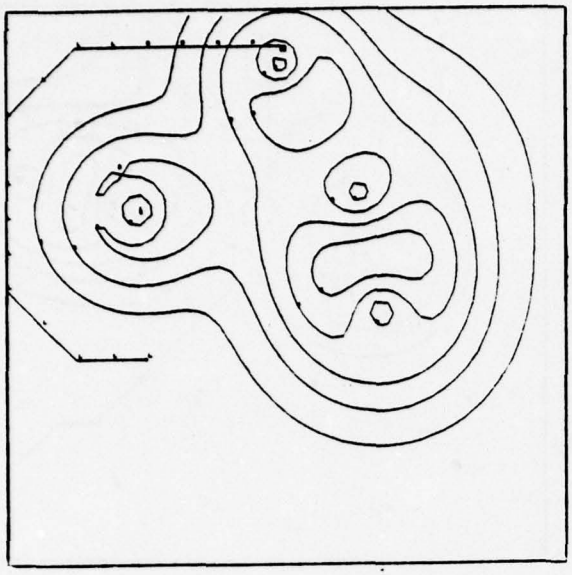
Problem 18



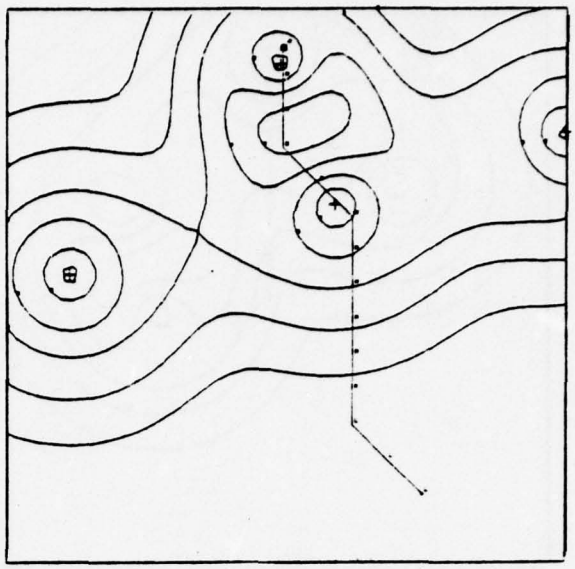
Problem 19



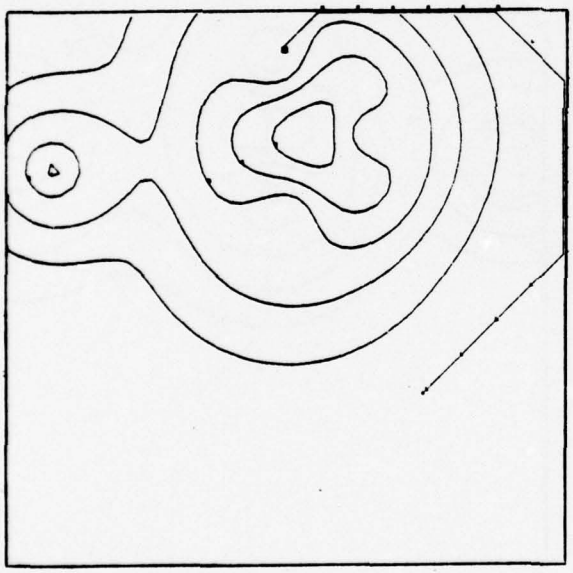
Problem 20



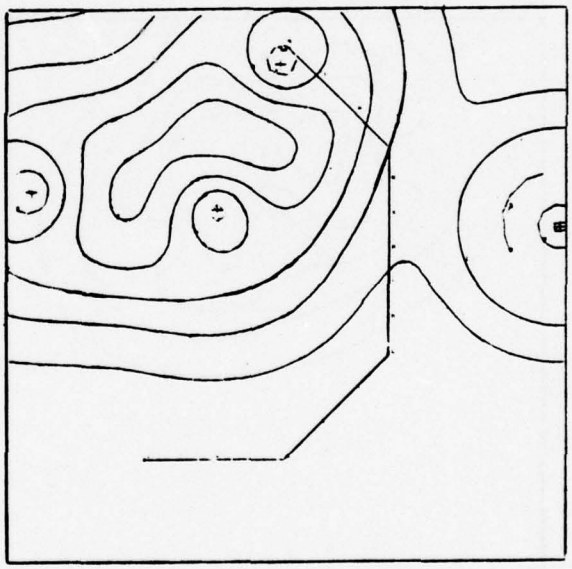
Problem 21



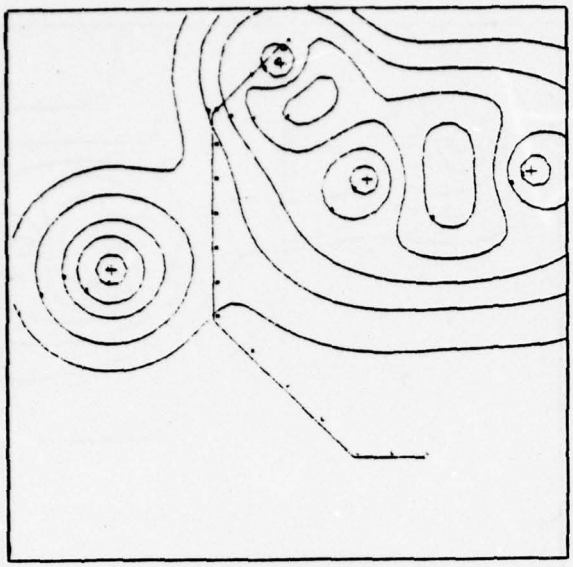
Problem 22



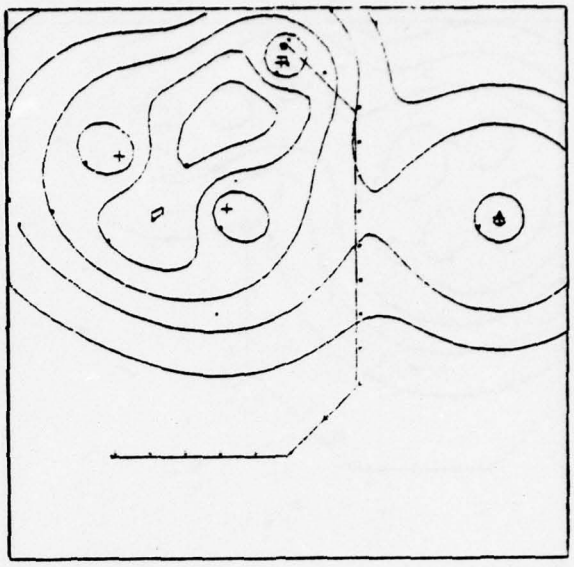
Problem 23



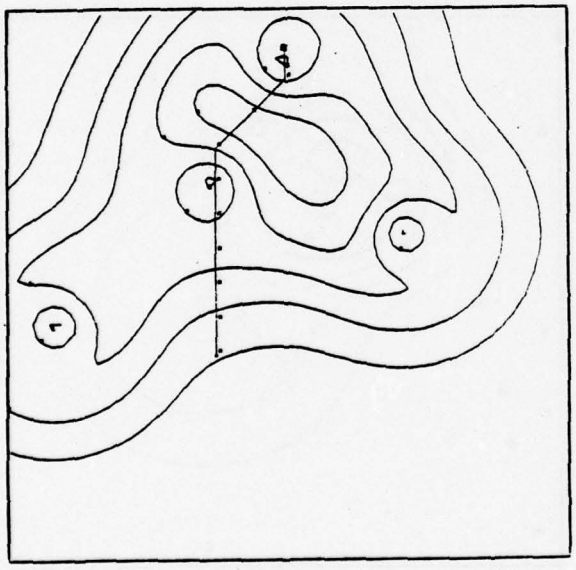
Problem 24



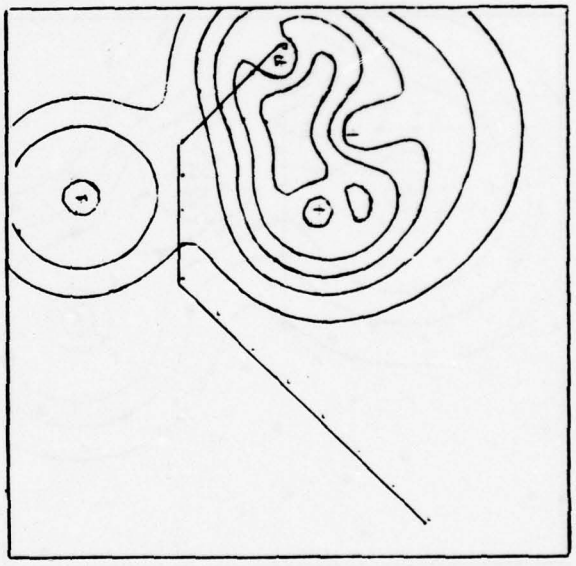
Problem 25



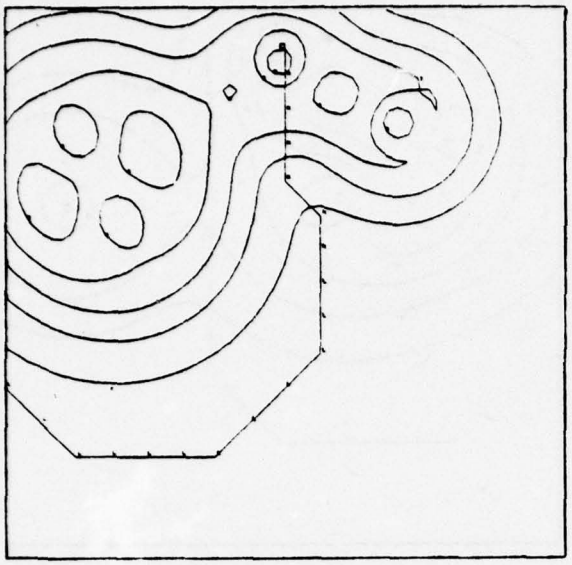
Problem 26



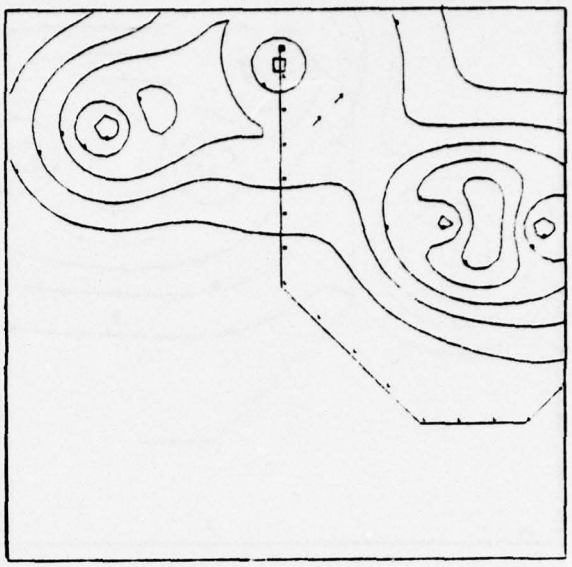
Problem 27



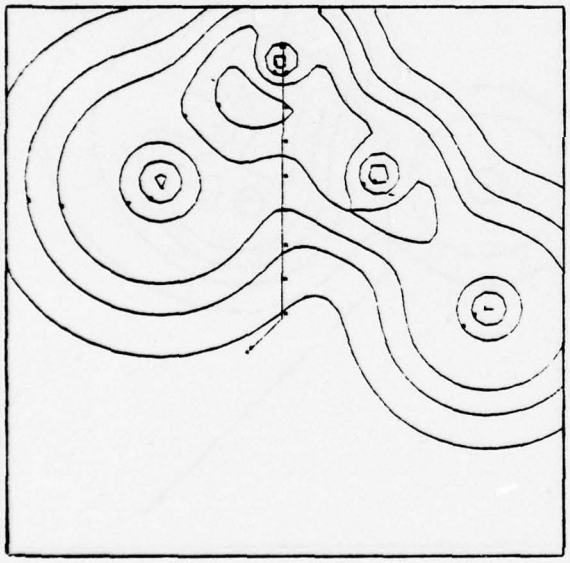
Problem 28



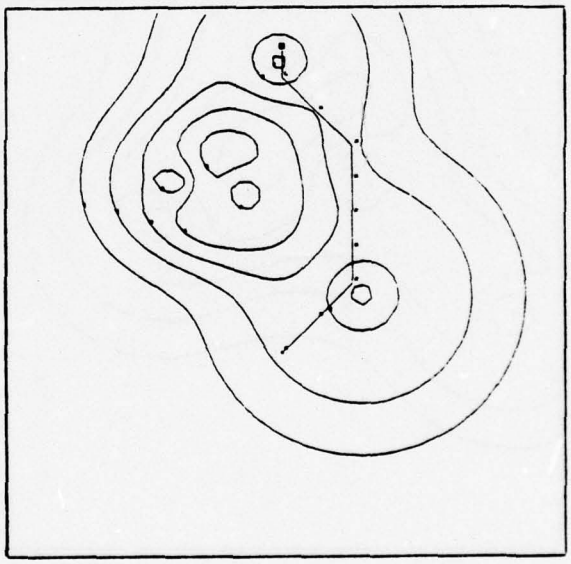
Problem 29



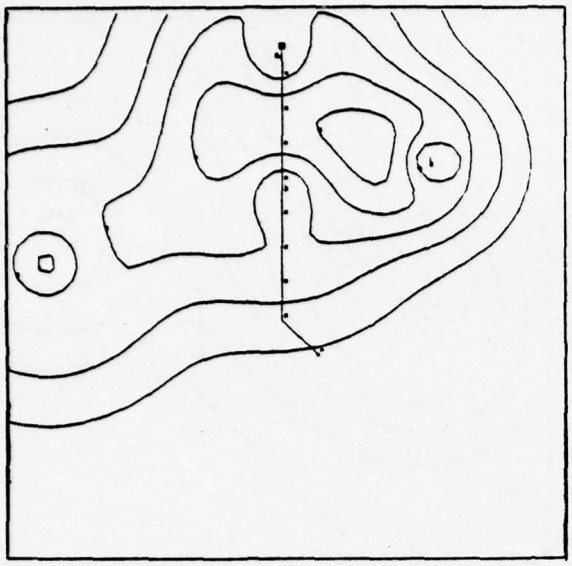
Problem 30



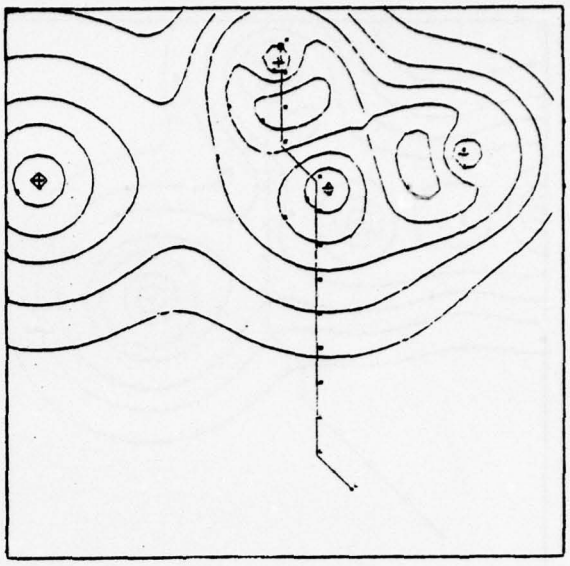
Problem 31



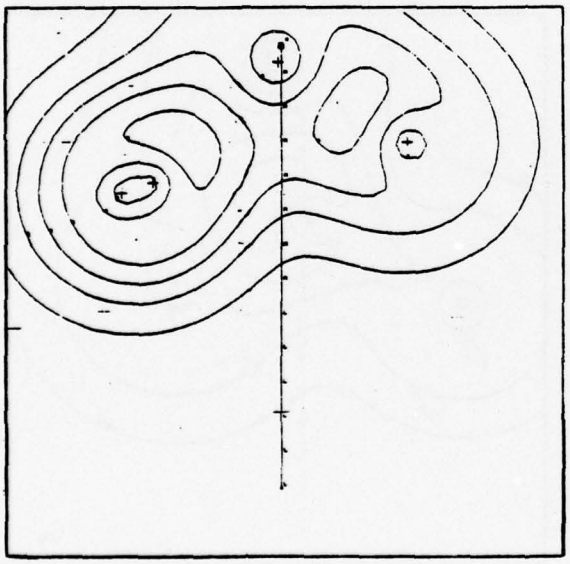
Problem 32



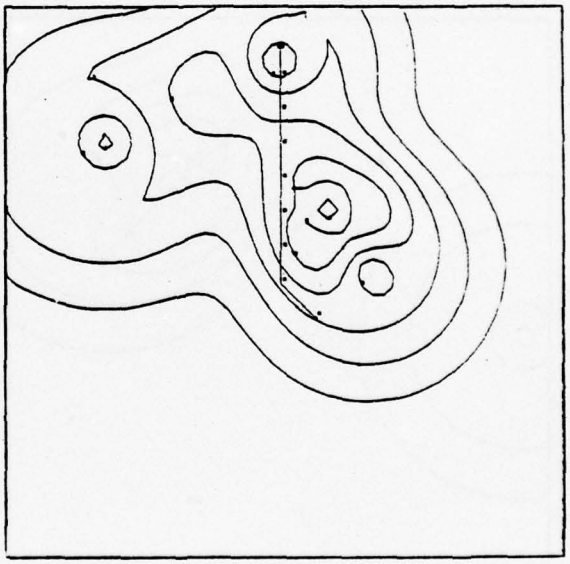
Problem 33



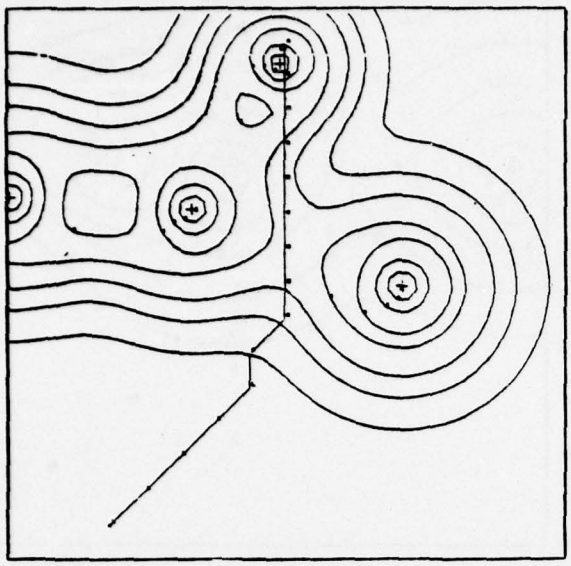
Problem 34



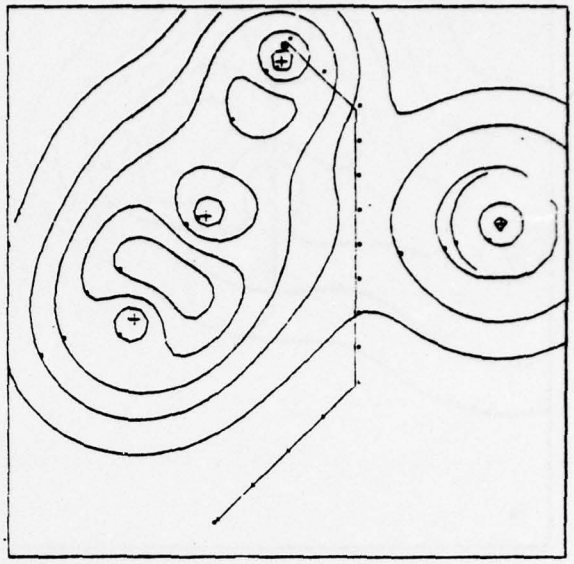
Problem 35



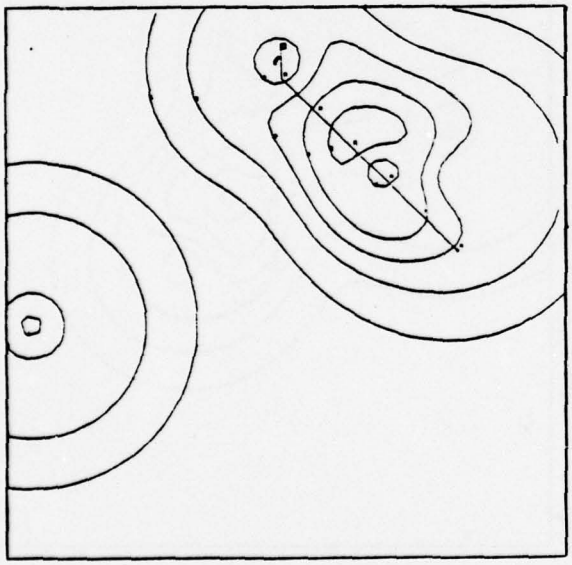
Problem 36



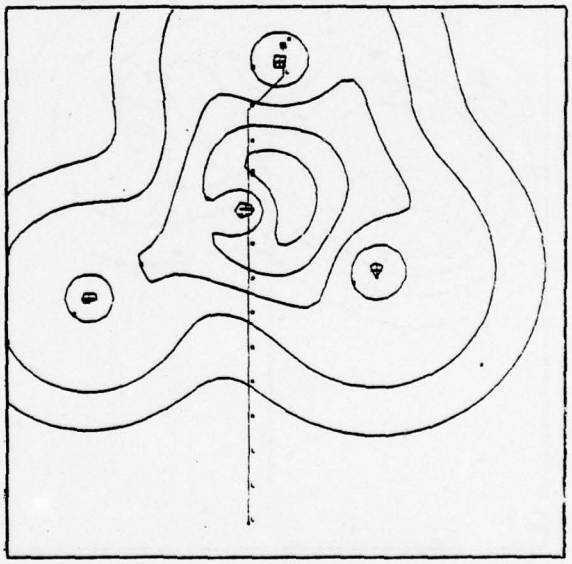
Problem 37



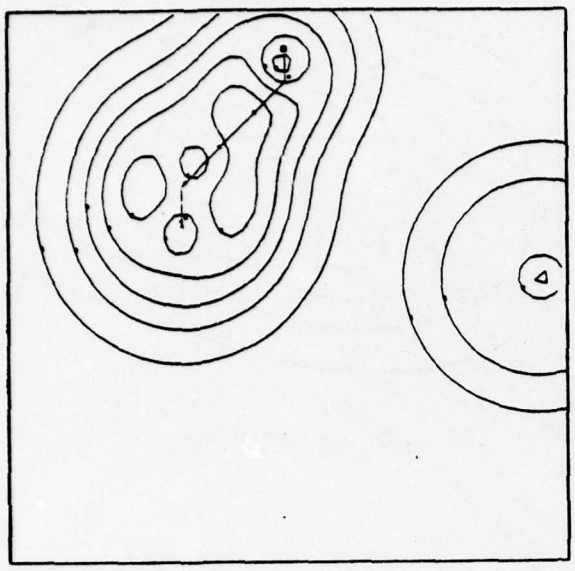
Problem 38



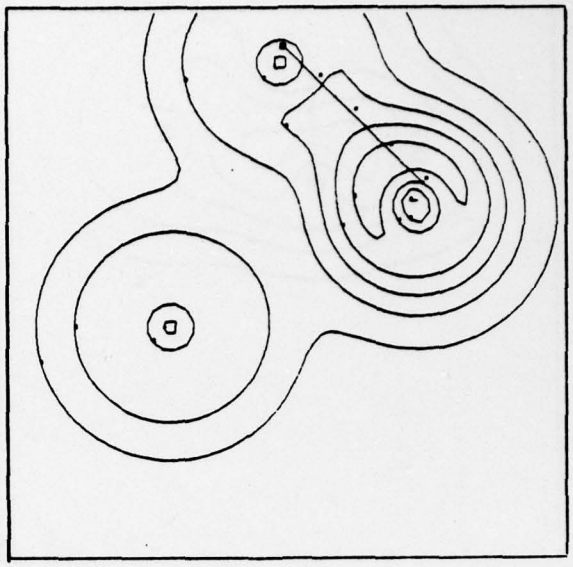
Problem 39



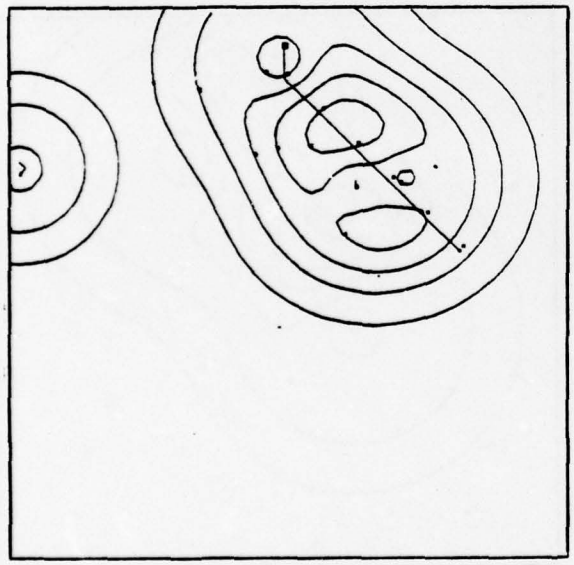
Problem 40



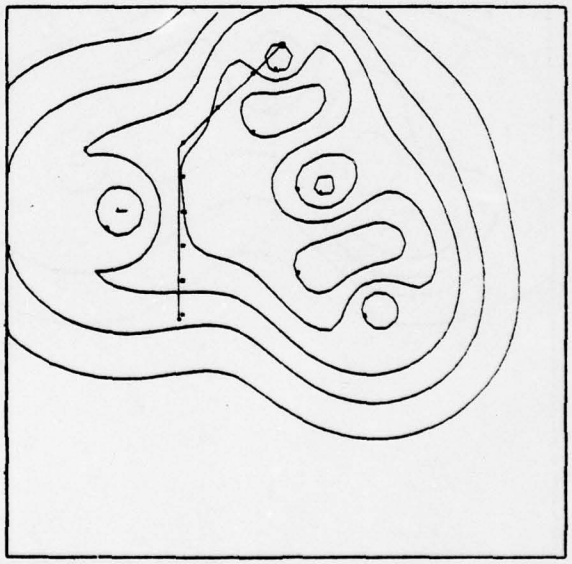
Problem 41



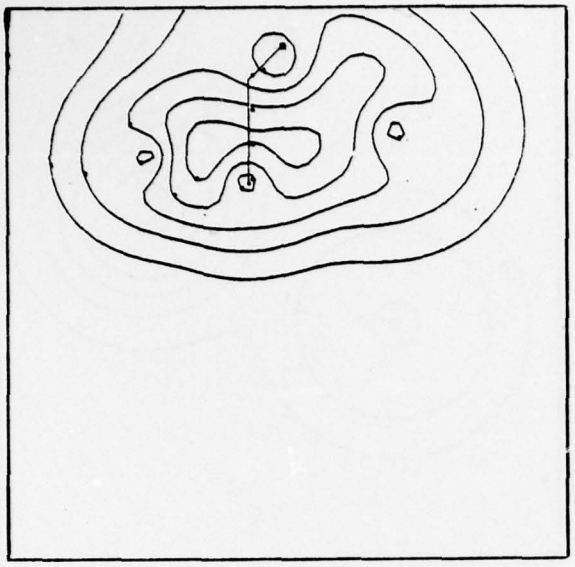
Problem 42



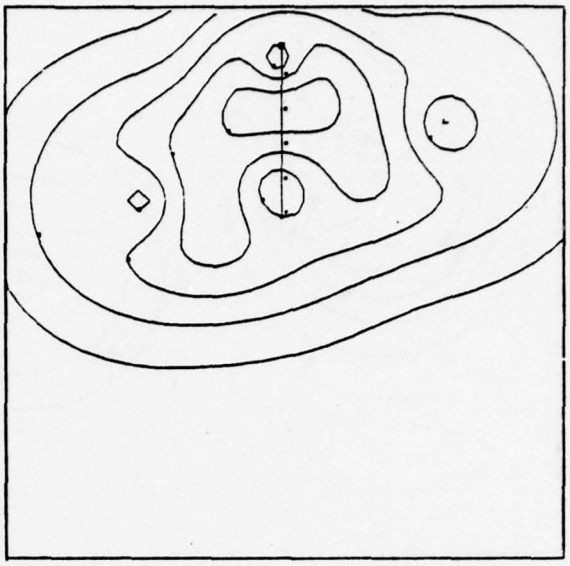
Problem 43



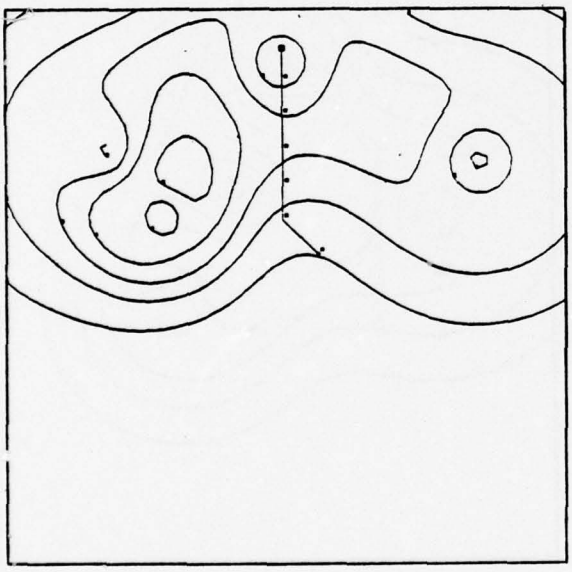
Problem 44



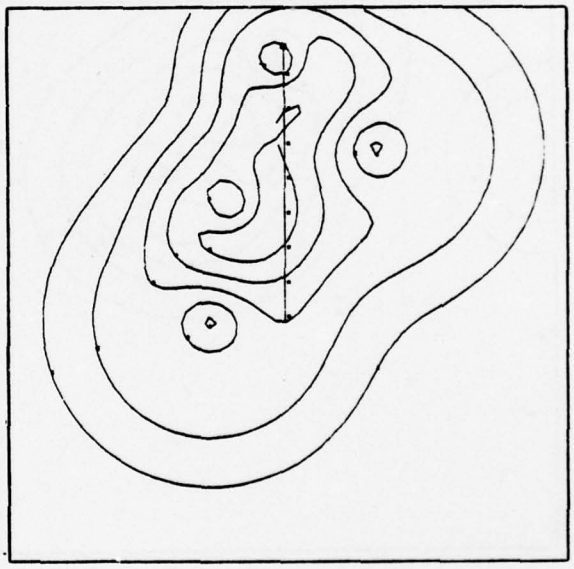
Problem 45



Problem 46



Problem 47



Problem 48

APPENDIX D

DYNAMIC PROGRAMMING OPTIMAL PATH SOLUTION

APPENDIX D: DYNAMIC PROGRAMMING OPTIMAL PATH SOLUTION

The experiment required best (or "answer") air strike paths (in the sense of optimal with respect to the utility function) in order to evaluate the operator estimates of optimal paths. Dynamic programming was selected as a convenient tool for obtaining solutions to this optimization problem. Specifically, a grid-oriented dynamic programming method was selected for the optimal solutions; its adaptation to the study is explained below.

1. Background

A detection rate field is generated by sensor performance models, and the task is to transit this field arriving at a target from a given starting point in such a way that the utility criterion function is maximized. Recall that the utility criterion function is a function of fuel remaining upon arrival at the target and the probability of being detected along a path.

The following restrictions were used to simplify the optimization problem.

1. The entire region of interest is put on a 16 x 16 quadruled grid so that all possible trajectories go through the grid points. Each grid point is also called a node.

2. Transitions are allowed only from a grid point to one of its immediately neighboring grid points. In other words, only cardinal and diagonal transitions are allowed, as shown in Figure 60. The "legal" transitions are numbered from 1 to 8.

3. For each transition, only one of three pre-selected velocities is allowed.

4. The problem then is, given two nodes A and B, to find both a path that transits through the grid points from A to B and its component velocities such that the utility criterion function associated with the path is maximized. For convenience call B the target and A the base.

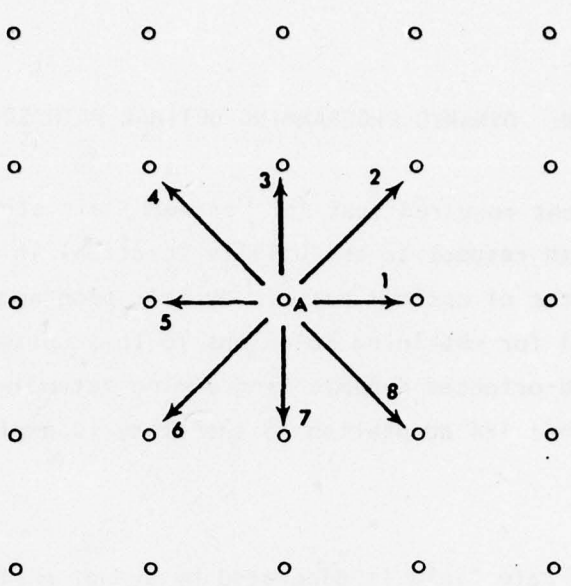


Figure 60. Allowable Air Strike Path Transitions.

This problem can be put in a form for solution by dynamic programming. Let us define a sequential decision making process by the following system:

The state of the system at stage k (k = 1, 2, 3, ...) is given by

$$X_k = (i, j, v, P_c, F) \tag{9}$$

where

(i, j) are the coordinates of the node we are currently at

v is the velocity for the transition ending at node (i, j)

P_c is the cumulative probability of not being detected by the sensor field for optimal path starting at i, j) and ending at B

F is the total fuel consumption for optimal path starting at (i, j) and ending at the target B

The decision D_k to be made at stage k is one of the eight allowable transitions defined previously and the velocity for the selected transition. Let $r(X_k, D_k)$ be the return (or utility) associated with the state X_k and decision D_k at stage k. Also let g be the criterion function for the returns from a sequence of stages.

Since an optimal path going from A to B is an optimal path going from B to A and vice versa, we will formulate the problem in the following manner:

Starting with the system at state X_1 corresponding to node B (target) we want to find a sequence of decisions (paths and velocities $D_1, D_2, \ldots, D_N$ for some positive number N such that $D_1, D_2, \ldots, D_N$ takes the system to some state X_N corresponding to node A (base). In addition, the criterion function $g\,[r(X_N, D_N), r(X_{N-1}, D_{N-1}), \ldots, r(X_1, D_1)]$ is to be maximized over all possible sequences of decisions (paths and velocities) going from B to A. This decision sequence $D_1, D_2, \ldots, D_N$ defines an optimal path from B to A and is an optimal path from A to B.

If the criterion function g is separable and monotonic, (Reference 3), it can be decomposed into g_1 and g_2 in the following manner.

$$\max_{D_k,\, D_{k-1},\, \ldots,\, D_1} \left\{ g[r(X_k, D_k), r(X_{k-1}, D_{k-1}), \ldots, r(X_1, D_1)] \right\} \tag{10}$$

$$= \max_{D_k} g_1[r(X_k, D_k), \max_{D_{k-1},\, \ldots,\, D_1} g_2(r(X_{k-1}, D_{k-1}), \ldots, r(X_1, D_1))]$$

Using dynamic programming, we can solve the above recursive equation by backward recursion. We start at X_1 and select a decision sequence $D_1, D_2, \ldots, D_N$, one stage at a time, that maximizes g. At state X_N the optimal path is given by $D_1, D_2, \ldots, D_N$.

We would carry out as many stages as necessary in order to include all grid points in our solution. What this means is that regardless of which point we later choose as the base, we can retrieve from our solution an optimal path going from base to target.

The return function r in Eq. (10) is related to the utility function by

$$r(X_m, D_m) = U(F_m, (1 - Pc_m)) \tag{11}$$

where

$X_m = (i_m, j_m, r_m, Pc_m, F_m)$ as defined in Eq. (9) and U is as defined in Eq. (5) in Section II.C.5.

Since in Eq. (10) we are trying to maximize $r(X_k, D_k) = U(F_k, (1-P_{ck}))$ at node (i_k, j_k), the criterion function g is a 1st projection function.* It follows that the function g_1 is also a 1st projection function and we can arbitrarily define the function g_2. However, the validity of the decomposition in Eq. (10) hinges on g being separable and monotonic. The criterion function g being a 1st projection function implies that it is separable. Unfortunately, though it is not obvious at first, a few sample calculations show that g is not monotonic. In spite of this fact, by using Eq. (10) and applying dynamic programming, we still arrived at an optimal answer in most of the cases we tried. When the solution was not optimal, it was very nearly optimal. We decided to retain the utility function and the dynamic programming optimization procedure due to time limitations to get the experiment under way. The experimental results support our belief that meaningful data could still be obtained, even though strictly speaking, our utility function may not (under certain conditions) be optimizable by dynamic programming.

2. Description of the Algorithm

We have just discussed the mathematical description of our optimal path problem. Now we desire a procedure for arriving at a solution. Let us go back to the grid we have laid over the entire region of interest. Picture our target B (destination) as being located on a grid point. Our problem then becomes, if we pick a starting point A (base), how to find an optimal path going to B. Using the dynamic programming approach, we would solve for an optimal path for each starting point on the grid.

Basically the method involves backward recursion, where we start at the target B and solve for optimal paths for all the immediate neighboring points of B that have allowable transitions into B. Then we go to the next layer of neighboring points and find their optimal paths. We propagate in this manner by going to successive layers of points until all points in the region are covered. This is a multiple pass method because we will reiterate until a solution is converged to. However, for practical reasons we will

---

*In general, the i^{th} projection function f_i is defined as

$$f_i(y_1, y_2, y_3, \ldots y_i, \ldots) = y_i$$

stop after the tenth iteration even if convergence has not occurred. Our experience shows us that after ten iterations the solution is near optimal.

A simplified flowchart describing this method is given in Figure 61. We start at the target B and define the first layer of points surrounding the target to be those points from which the target can be reached in one transition (see Figure 62). For each point in this layer we find an optimal path to the target by examining the neighbors of this point and selecting a neighbor and a transition velocity such that a transition thereto yields an optimal path. When we are done we proceed to the second layer of points surrounding the target. This is the set of outside points from which the first layer can be reached in one transition. For each point in the second layer (for example, see point C in Figure 63), we again select a neighbor and a transition velocity such that the transition thereto, plus the already found optimal path for that neighbor, constitutes an optimal path for the point. Then we go to the third layer of points around the target, and so on until the entire region is covered by our layers. This is the completion of a pass, and at this point we check to see if we have defined any new optimal path during the pass. If so, we go back to the first layer of points to start another pass. We would have converged to a solution if no change was made to the optimal paths during a pass.

To retrieve an optimal path, we go to a grid point corresponding to the start of the air strike, and obtain an optimal transition and velocity. This information tells us which grid point is the next point on our optimal path. Then we go to this next grid point and again obtain an optimal transition and velocity. By repeating this process, the successive transitions and velocities so obtained define an optimal path and we stop when we have arrived at the target.

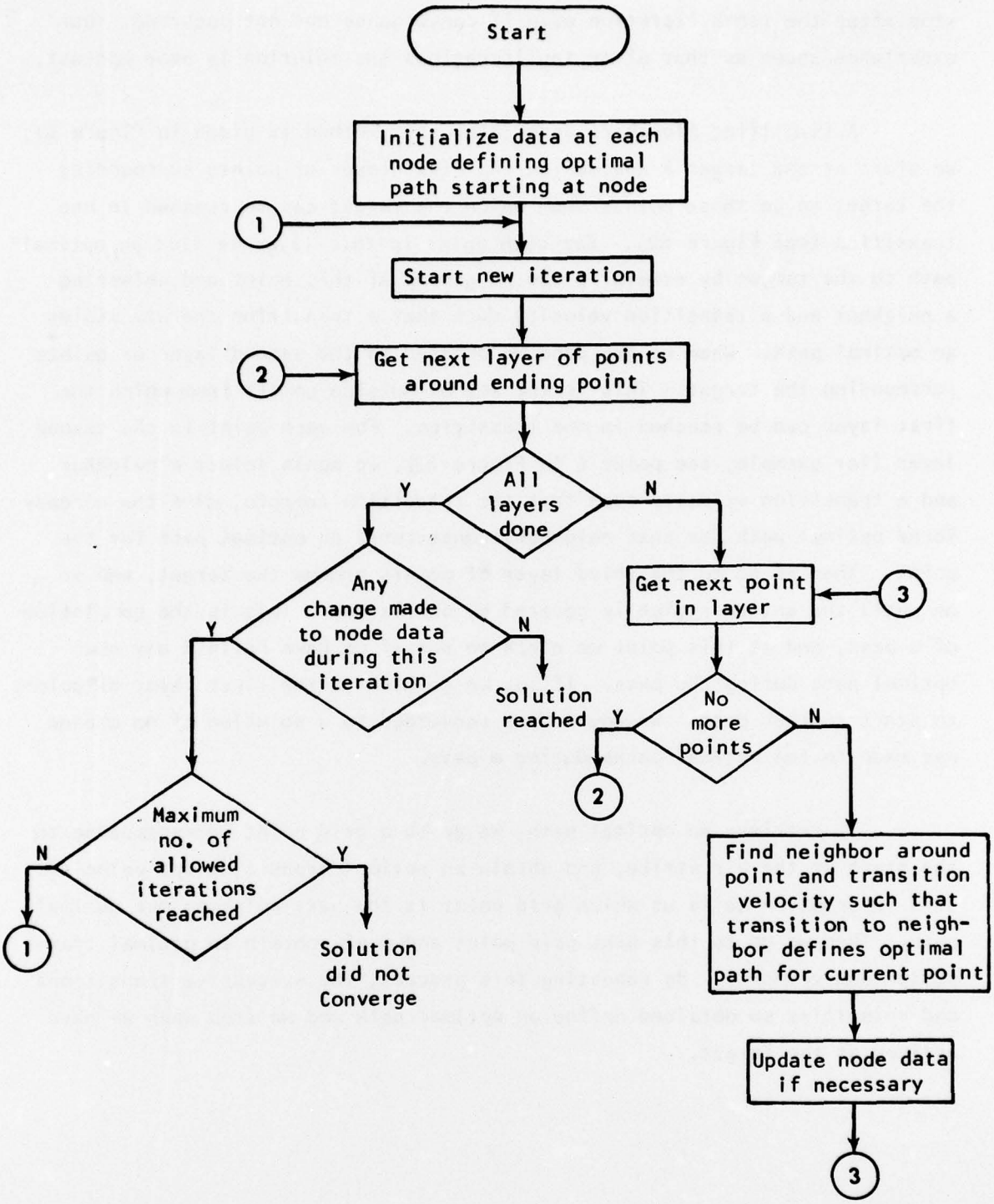


Figure 61. Flowchart for Dynamic Programming Procedure.

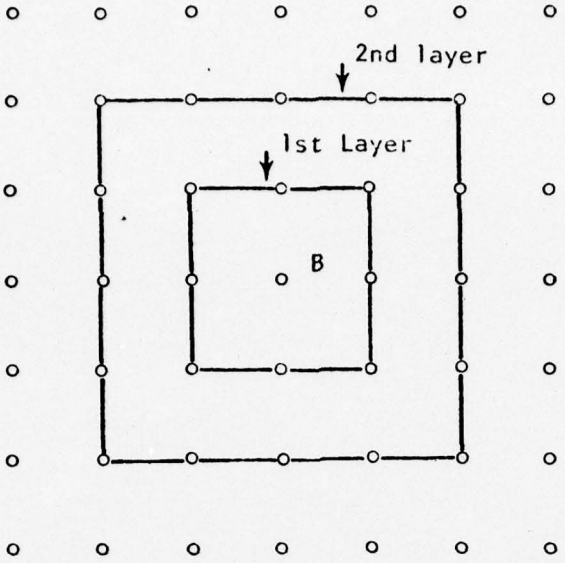


Figure 62. Layers of Points Around Target (B).

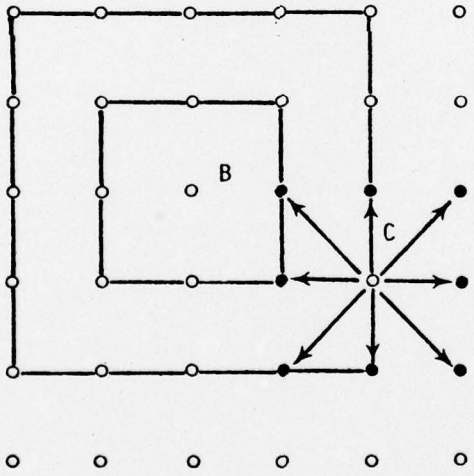


Figure 63. Potential Optimal Nodes. (From point C, we are to decide which of the filled-in nodes is to be on its optimal path to B. An arrow indicates a candidate transition.)

DISTRIBUTION

Organization	No. of Copies
Director, Engineering Psychology Program Psychological Sciences Division Office of Naval Research 800 N. Quincy Street Arlington, VA. 22217 ATTN: Martin A. Tolcott Code 455 Ref: Contract Number N00014-75-C-0811	5
E.J. Sosknowski Defense Contract Administrative Services 11099 South La Cienega Boulevard Los Angeles, CA. 90045	1
Director Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20375	6
Office of Naval Research Department of the Navy Arlington, VA. 22217 ATTN: Code 1021P	6
Defense Documentation Center Building 5, Cameron Station Alexandria, VA. 22314	12
Director, ONR Branch Office 1030 East Green Street Pasadena, CA. 91106	1
Lt. Col. Henry L. Taylor, USAF OAD(E&LS) ODDR&E Pentagon, Rm 3D129 Washington, D.C. 20301	1
Captain R. Granum Office of Assistant Secretary of Defense (Intelligence), Pentagon Washington, D.C. 20301	1
Dr. Robert Young Director Human Resources Research Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA. 22209	1

Organization	No. of Copies
Personnel Logistics Plans, OP987P10 Office of the Chief of Naval Operations Department of the Navy Washington, D.C. 20350	1
Dr. A.L. Slafkosky Scientific Advisor Commandant of the Marine Corps Code RD-1 Washington, D.C. 20380	1
Assistant Chief for Technology, Code 200 Office of Naval Research 800 N. Quincy Street Arlington, VA. 22217	1
Fleet Analysis and Support Division, Code 230 Office of Naval Research 800 N. Quincy Street Arlington, VA. 22217	1
Naval Analysis Programs, Code 431 Office of Naval Research 800 N. Quincy Street Arlington, VA. 22217	1
Operations Research Program, Code 434 Office of Naval Research 800 N. Quincy Street Arlington, VA. 22217	1
Statistics and Probability Program, Code 436 Office of Naval Research 800 N. Quincy Street Arlington, VA. 22217	1
Information Systems Program, Code 437 Office of Naval Research 800 N. Quincy Street Arlington, VA. 22217	1
Dr. Fred Muckler Manned Systems Design, Code 311 Navy Personnel Research and Development Center San Diego, CA. 92152	1
Dr. Allen C. Miller Management Analysis Company of Palo Alto 1900 Embarcadero Road, Suite 207 Palo Alto, CA. 94303	1

Organization	No. of Copies
LCDR M. O'Bar, Code 9305 Navy Personnel Research and Development Center San Diego, CA. 92152	1
Navy Personnel Research and Development Center Management Support Department, Code 210 San Diego, CA. 92152	1
Naval Electronics Systems Command Human Factors Engineering Branch Code 4701 Washington, D.C. 20360	1
Mr. Arnold Rubinstein Naval Material Command, NAVMAT 0344 Department of the Navy Washington, D.C. 20360	1
Mr. John Silva Head, Human Factors Division Naval Ocean Systems Center San Diego, CA. 92152	1
Dr. Jesse Orlansky Institute for Defense Analyses 400 Army-Navy Drive Arlington, VA. 22202	1
Ralph M. Tucker Martin Marietta Aerospace Mail Stop 8105 Denver Division P.O. Box 179 Denver, CO. 80201	1
Human Factors Dept., Code N215 Naval Training Equipment Center Orlando, FL. 32813	1
Dr. Alfred F. Smode Training Analysis & Evaluation Group Naval Training Equipment Center Code N-00T Orlando, Fl. 32813	1
Dr. Gary Poock Operations Research Department Naval Postgraduate School Monterey, CA. 93940	1

Organization	No. of Copies
Dr. Joseph Zeidner Dir., Organization and Systems Research Laboratory U.S. Army Research Institute 1300 Wilson Boulevard Arlington, VA. 22209	1
Dr. Donald A Topmiller Chief, Systems Effect. Branch Human Engineering Division Wright Patterson AFB, OH. 45433	1
Dr. H.W. Sinaiko Smithsonian Institution 801 N. Pitt Street Alexandria, VA. 22314	1
Dr. Gary Lucas System Planning Corporation 1500 Wilson Boulevard Arlington, VA. 22209	1
Dr. C. Peterson Decisions and Designs, Inc. 8400 Westpark Drive, Suite 6-0 McLean, VA. 22101	1
Mr. Victor Monteleon Code 230 Naval Ocean Systems Center San Diego, CA. 92152	1
Navy C3 Architecture Division, OP-943 Office of the Chief of Naval Operations ATTN: Capt. Robert Amman 3801 Nebraska Avenue, N.W. Washington, D.C. 20390	1
Mr. L.A. Aarons R&D Plans Division Office of the Chief of Naval Operations OP-987C Washington, D.C. 20350	1
Commander, Naval Electronics Systems Command Command and Control Division, Code 530 Washington, D.C. 20360	1
Commander, Naval Electronics Systems Command C^3 Project Office PME 108-1 Washington, D.C. 20360	1

Organization	No. of Copies
LCDR Charles Theisen NAVAIRDEVCEN, Code 4024 Warminster, PA 18974	1
Mr. Samuel Epstein Analytics Inc. 1405 Colshire Drive McLean, VA. 22101	1
Dr. Bertram Spector CACI, Inc. - Federal 1815 N. Fort Myer Drive Arlington, VA. 22209	1
Mr. Harold Crane CTEC, Inc. 7777 Leesburg Pike Falls Church, VA. 22043	1
Dr. Robert Andrews Organizations & Systems Research Lab U.S. Army Research Lab 1300 Wilson Boulevard Arlington, VA. 22209	1
Mr. George Pugh Decision-Science Applications Inc. 1401 Wilson Blvd. Arlington, VA 22209	1
Dr. Amos Freedy Perceptronics, Inc. 6271 Variel Avenue Woodland Hills, CA. 91364	1
Mr. Lee Merkhofer Stanford Research Institute Decision Analysis Group Menlo Park, CA. 94025	1
Mr. Victor Rowney Stanford Research Institute Naval Warfare Research Center Menlo Park, CA. 94025	1
Dr. John Shore Code 5403 Communications Sciences Division Naval Research Laboratory Washington, D.C. 20375	1

Organization	No. of Copies
Dr. H.L. Morgan University of Pennsylvania Wharton School Philadelphia, PA. 19174	1
M.L. Matursky NAVAIRDEVCEN, Code 5424 Warminster, PA. 18974	1
Lt. Col. David Dianich HQS Tactical Air Command Langley AFB, VA. 22065	1
Commander, Naval Electronics Systems Command ELEX-03 Washington, D.C. 20360	1
Dr. Chantee Lewis Management Department Naval War College Newport, R.I. 02840	1
CDR Richard Schlaff NIPSSA Hoffman Bldg. #1 2461 Eisenhower Avenue Alexandria, VA. 22331	1